Slabicki, Mariusz; Premsankar, Gopika; Di Francesco, Mario

Adaptive configuration of LoRa networks for dense IoT deployments

Published in:
IEEE/IFIP Network Operations and Management Symposium

DOI:
10.1109/NOMS.2018.8406255

Published: 01/01/2018

Document Version
Peer reviewed version

Please cite the original version:
Slabicki, M., Premsankar, G., & Di Francesco, M. (2018). Adaptive configuration of LoRa networks for dense IoT deployments. In IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018 (pp. 1-9). (IEEE/IFIP Network Operations and Management Symposium). IEEE. https://doi.org/10.1109/NOMS.2018.8406255

Adaptive Configuration of LoRa Networks for Dense IoT Deployments

Mariusz Slabicki, Gopika Premsankar, Mario Di Francesco
Department of Computer Science, Aalto University, Finland
Email: {mariusz.slabicki, gopika.premsankar, mario.di.francesco}@aalto.fi

***Abstract*—Large-scale Internet of Things (IoT) deployments demand long-range wireless communications, especially in urban and metropolitan areas. LoRa is one of the most promising technologies in this context due to its simplicity and flexibility. Indeed, deploying LoRa networks in dense IoT scenarios must achieve two main goals: efficient communications among a large number of devices and resilience against dynamic channel conditions due to demanding environmental settings (e.g., the presence of many buildings). This work investigates adaptive mechanisms to configure the communication parameters of LoRa networks in dense IoT scenarios. To this end, we develop FLoRa, an open-source framework for end-to-end LoRa simulations in OMNeT++. We then implement and evaluate the Adaptive Data Rate (ADR) mechanism built into LoRa to dynamically manage link parameters for scalable and efficient network operations. Extensive simulations show that ADR is effective in increasing the network delivery ratio under stable channel conditions, while keeping the energy consumption low. Our results also show that the performance of ADR is severely affected by a highly-varying wireless channel. We thereby propose an improved version of the original ADR mechanism to cope with variable channel conditions. Our proposed solution significantly increases both the reliability and the energy efficiency of communications over a noisy channel, almost irrespective of the network size. Finally, we show that the delivery ratio of very dense networks can be further improved by using a network-aware approach, wherein the link parameters are configured based on the global knowledge of the network.**

***Index Terms*—LoRa, configuration management, performance evaluation, adaptive data rate, reliability, energy efficiency**

I. Introduction

Several communication technologies and standards have been proposed for Low-Power Wide Area Networks (LPWANs) [1, 2]. Among them, LoRa [3, 4] has gained momentum due to its low complexity and the use of unlicensed Industrial Scientific and Medical (ISM) bands. In fact, there are already several LoRa service providers, including The Things Network [5], an open and community-driven Internet of Things (IoT) network. However, the management of LoRa networks for IoT deployments faces several challenges, including: a large and highly-varying number of nodes; diverse wireless scenarios characterized by demanding environmental factors (e.g., dense urban settings with many buildings); interference due to other co-located networks operating on the same unlicensed frequency bands. These challenges may compromise scalability, eventually affecting the reliability of services running on top of IoT deployments [6]. One option to address such challenges is to dynamically adapt the operating parameters of wireless communications in the network.

Although many recent works have characterized the communication performance of LoRa networks [7–10], they have not thoroughly investigated the impact of dynamic management of communication parameters. Indeed, this article targets adaptive configuration of LoRa networks for scalable IoT deployments. Such an adaptive configuration can be achieved with either a link-based or a network-aware approach. Link-based schemes configure the communication parameters independently for each wireless link between nodes. On the other hand, network-aware approaches adapt these parameters for a certain link by leveraging a global knowledge of the nodes in the network. In this regard, our work evaluates both approaches to configure the transmission parameters at the physical layer of LoRa to improve both reliability and energy efficiency in dense IoT scenarios.

The major contributions of this work are the following. First, we develop FLoRa, an open-source framework for LoRa simulations in OMNeT++ [11]. FLoRa implements the physical and medium access control layers of LoRa, supports bi-directional communications and allows end-to-end simulations including the backhaul network. Second, we evaluate the performance of link-based adaptation in LoRa using the Adaptive Data Rate (ADR) mechanism. We conduct extensive simulations and show that ADR is effective in increasing the delivery ratio for networks with stable channel conditions, while keeping the energy consumption low. We also show that the performance of ADR is severely affected by a highly-varying wireless channel. We thereby propose an improved version of the original ADR mechanism to cope with variable channel conditions. Our proposed solution significantly increases both the reliability and the energy efficiency of communications over a noisy channel, almost irrespective of the network size. Finally, we show that the delivery ratio of very dense networks can be further improved by using a network-aware approach, wherein the link parameters are configured based on the global knowledge of the network (e.g., the location of the devices).

The rest of the article is organized as follows. Section II provides background on LoRa. Section III describes the implementation of our simulator. Section IV discusses adaptive communication management in LoRa networks. Section V presents the simulation setup and the obtained results. Section VI reviews the state of the art. Finally, Section VII provides some concluding remarks.

II. Overview of LoRa

A LoRa network relies on two components, namely, LoRa and LoRaWAN, each corresponding to a different layer of the protocol stack. LoRa itself is a proprietary physical layer developed by Semtech Corporation. LoRaWAN, on the other hand, is described in an open specification developed by the LoRa Alliance [3]. LoRaWAN constitutes the medium access control (MAC) and network layers in LoRa networks.

The LoRa physical layer enables long-distance, low-power communication and operates in the unlicensed sub-GHz ISM band. LoRa utilizes chirp spread spectrum modulation to encode an input signal into chirp pulses spread over a wide spectrum [3]. This technique enables long-distance communication, even though this results in a low data rate. Each LoRa transmission is characterized by five parameters: spreading factor, transmission power, code rate, center frequency and bandwidth [6]. These parameters affect the communication range, the data rate, the robustness to interference or noise, and the ability of a receiver to decode the signal. The available values for each parameter depend on the region where LoRa devices are deployed [12]. The *spreading factor* is the ratio between the data symbol rate and chirp rate. The configuration of the spreading factor allows tuning the data rate and the reachable distance. In fact, the data rate is lower at higher spreading factors, but the communication range is higher. Choosing different spreading factors also enables orthogonal signals, implying that a receiver can successfully receive distinct signals sent over a given channel at the same time [3]. The *transmission power* can be configured based on the region and the bandwidth used for transmissions. The *code rate* is the forward error correction rate, and it affects the airtime of packet transmissions. The *center frequency* depends on the ISM band used in a particular region. Finally, the *bandwidth* influences the data rate of transmissions.

The LoRaWAN specification [4] defines an open standard for the network protocols and the system architecture of LoRa networks. LoRaWAN relies on an ALOHA-based MAC protocol [13], which reduces the complexity of end-devices. The network architecture consists of a hierarchical topology, wherein LoRa nodes communicate with gateways over the LoRa physical layer. A node is not associated with a particular gateway; instead, all gateways within the range of a transmitter can receive messages. The gateways simply relay received messages to a central network server. The communication between gateways and the network server takes place over the standard Internet Protocol (IP). The central network server manages the network by processing the incoming messages, filtering duplicate packets, forwarding messages to application servers, and sending responses to nodes through a single designated gateway.

III. A Framework for LoRa simulations

We developed FLoRa[1] (Framework for LoRa), a simulation tool based on the OMNeT++ [11] discrete event simulator and its INET framework [14]. FLoRa is open source and includes modules that simulate the LoRa physical layer, the LoRaWAN MAC protocol as well as network elements including gateways and network servers. Moreover, FLoRa includes a module to characterize the energy consumption of LoRa end devices. This section details the implementation of FLoRa. We first describe the characterization of the LoRa physical layer, then discuss the adopted energy model. We also present the key features of the network elements and the architecture supported by the simulator.

[1] http://flora.aalto.fi/

TABLE I: Parameters of the propagation model [6, 16].

Scenario	d_0 [m]	$\overline{PL}(d_0)$ [dB]	n	σ [dB]
Urban	40	127.41	2.08	3.57
Sub-urban	1000	128.95	2.32	7.08

A. LoRa Links

FLoRa allows to configure all the transmission parameters in the LoRa physical layer: spreading factor, center frequency, bandwidth, code rate, and transmission power. These parameters determine the communication range and the occurrence of collisions. In particular, a LoRa transmission is successful if the received power is greater than the receiver sensitivity. The received power depends on the transmission power and the losses due to signal attenuation and shadowing. This is modeled using the well-known log-distance path loss model with shadowing [15], which calculates the path loss based on the distance between the transmitter and receiver as follows:

$$PL(d) = \overline{PL}(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma \qquad (1)$$

where $\overline{PL}(d_0)$ is the mean path loss for distance d_0, n is the path loss exponent, and X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ.

FLoRa supports both urban and sub-urban environments. Table I shows the default path loss parameters ($\overline{PL}(d_0)$, d_0, σ and n) used in the two cases. These parameters have been derived from the measurements carried out in [6] and [16]. Specifically, the measurements in [6] correspond to a built-up urban environment wherein deployments are partially indoors. The measurements in [16], instead, correspond to a sub-urban environment with only a few tall buildings. Thus, the communication range in sub-urban areas [16] is higher than the one in urban areas [6].

A successful transmission also depends on whether LoRa transmissions interfere with each other or not. To this end, we use the collision model proposed in [6]. The main assumption is that two transmissions in orthogonal channels (for instance, transmissions with different spreading factors) do not collide. If two messages are in non-orthogonal channels, a collision occurs when they overlap in time. Furthermore, the collision model includes the capture effect as experimentally characterized in [6]. Accordingly, the stronger of two colliding signals is decoded, provided that the power difference between the signals is more than 6 dBm and at least 5 symbols in the

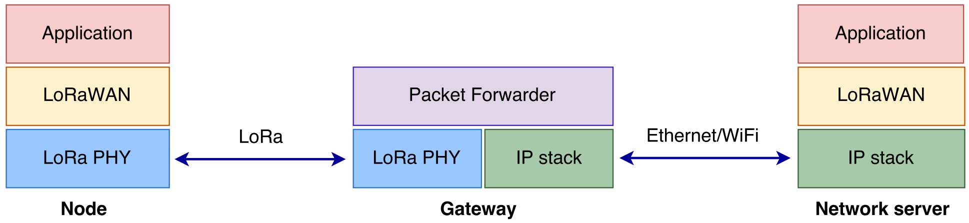


Fig. 1: Modules available in FLoRa and the corresponding protocol stack.

preamble are detected. We validated the communication model of our simulator against the experimental results from [6, 16] and verified that it yields results very close to those presented in the two articles.

B. Energy Consumer Module

The energy expenditure is modeled by a state-based *energy consumer* module, wherein the energy consumed depends on the amount of time spent by the LoRa radio in a particular state. The three main states of a LoRa radio are transmit, receive and sleep. The radio is switched to sleep mode after transmitting or receiving a frame [17]. The energy consumed in the transmit state depends on the transmission power level. The values of the instantaneous current for each transmission power level are obtained from [6]. The current drawn during the receive and sleep modes are derived from the Semtech SX1272/73 datasheet [17] with a supply voltage of 3.3 V.

C. Network Elements

FLoRa enables end-to-end network simulations by modeling LoRa nodes, gateways and network servers. Figure 1 shows the relevant modules and the related protocol stack. The gateway is able to receive LoRa transmissions from nodes on multiple channels simultaneously, in compliance with the LoRaWAN specifications [4]. The gateway and the network server communicate over IP. The physical layer between the gateway and the network server can be realized with existing INET modules such as Ethernet and WiFi links. Similarly, delays in the backhaul network can be described by an appropriate configuration of the relevant link parameters. A network can contain multiple gateways; the network server filters out duplicate packets from multiple gateways and sends downlink data to a node through the gateway with the highest link quality indicator. The network server also implements the management algorithms described in the next section.

IV. Adaptive Configuration of LoRa Networks

LoRa networks are expected to support a large number of devices exchanging data over the LoRa physical layer. Managing the transmission parameters of the physical layer (described in Section II) plays an important role in determining the capacity and scalability of LoRa networks. To this end, the LoRaWAN specification describes a link-based Adaptive Data Rate (ADR) mechanism [4], which dynamically modifies the transmission parameters for links between nodes and gateways. This section first describes the features of ADR. We then propose an improved version of the ADR scheme to achieve better performance under variable channel conditions. Finally, we describe a network-aware approach, wherein a global knowledge on the network is used to adapt the transmission parameters.

A. Adaptive Data Rate

ADR is designed to efficiently set the data rate and the transmission power of static nodes with two main goals: increase the overall capacity of the network and maximize the battery life of the nodes [3, 4]. ADR achieves this by estimating a link budget, which is the sum of all gains and losses in each wireless link between a node and a gateway. For instance, a node located close to a gateway can transmit data with a low spreading factor so as to lower the message transmission time. This allows other nodes to utilize the available channel for other transmissions [3]. Battery life is increased by dynamically assigning the transmission power of a node based on its distance to the gateway.

The ADR mechanism runs asynchronously at the LoRa node and at the network server. Most of the complexity in ADR is assigned to the network server, with the goal to keep the nodes as simple as possible. The ADR algorithm on the node is specified by the LoRa Alliance [4], whereas the algorithm on the network server is defined by the network operator. While the ADR algorithm at the network server can decrease the spreading factor (SF) and modify the transmission power (TP), the algorithm at the node can only increase the SF.

The part of ADR running at the node (ADR-NODE) is described in Algorithm 1. Its main goal is to increase the SF (thereby reducing the data rate) if uplink transmissions cannot reach the gateway. If a downlink frame is not received within a configurable number of frames, the node increases the SF of the subsequent uplink frame. This increases the transmission range and, thus, also the probability of reaching a gateway.

The part of ADR running at the network server[2], referred to as ADR-NET, is described in Algorithm 2. The algorithm allows the network server to change the TP and the SF for the uplink data transmissions of end nodes. To this end, the network server estimates the link budget of each node by using the SNR of received frames. The transmission parameters are then estimated based on the knowledge of the minimum SNR (SNR_{req}) required for demodulation, which is adjusted by a

[2]We refer to the version of ADR running at the network server according to the implementation in the Things Network [18], which is based on the reference (yet not publicly-available) rate adaptation algorithm by Semtech.

Algorithm 1 ADR-NODE

$ADR\_ACK\_LIMIT \leftarrow 64$
$ADR\_ACK\_DELAY \leftarrow 32$
$ADR\_ACK\_CNT \leftarrow 0$
if uplink transmission **then**
    $ADR\_ACK\_CNT \leftarrow ADR\_ACK\_CNT + 1$
if $ADR\_ACK\_CNT == ADR\_ACK\_LIMIT$ **then**
    Request response from network server
if $ADR\_ACK\_CNT \geq ADR\_ACK\_LIMIT + ADR\_ACK\_DELAY$ **then**
    increase SF
if downlink transmission received **then**
    $ADR\_ACK\_CNT \leftarrow 0$

Algorithm 2 ADR-NET

1: $SNR_m \leftarrow max$(SNR of last 20 frames)
2: $SNR_{req} \leftarrow$ demodulation floor(current data rate)
3: $deviceMargin \leftarrow 10$
4: $SNR_{margin} \leftarrow (SNR_m - SNR_{req} - deviceMargin)$
5: $steps \leftarrow floor(SNR_{margin}/3)$
6: **while** steps > 0 **and** SF $> SF_{min}$ **do**
7:     $SF \leftarrow SF - 1$
8:     $steps \leftarrow steps - 1$
9: **while** steps > 0 **and** TP $> TP_{min}$ **do**
10:     $TP \leftarrow TP - 3$
11:     $steps \leftarrow steps - 1$
12: **while** steps < 0 **and** TP $< TP_{max}$ **do**
13:     $TP \leftarrow TP + 3$
14:     $steps \leftarrow steps + 1$
15: **end**

device-specific margin. The newly calculated parameters are communicated to the LoRa node through a downlink frame. The node uses the new parameters for future transmissions until otherwise instructed. Note that the network server does not increase the SF (i.e., it does not reduce the data rate), as this is done by the LoRa node through ADR-NODE.

B. Improving ADR-NET

The ADR-NET algorithm estimates the link quality by using the *maximum* SNR value from historical samples. This choice is ideal when there is no variability in the channel quality. However, taking the maximum value is an optimistic approach to estimate link quality when the physical channel conditions are variable, for instance, due to weather or moving obstacles between two communicating devices. Thus, we propose a simple modification, wherein the *max* operator in line 1 of Algorithm 2 is replaced with the *average* function. We refer to this new algorithm as *ADR+*. We propose such a change to solve the problem of high variability in fast-fading conditions by using a more conservative estimation of the wireless channel as shown in the next section.

TABLE II: Fraction of nodes assigned to each spreading factor.

Spreading factor	7	8	9	10	11	12
Percentage of nodes	45.6	25.5	14.6	7.4	4.6	2.3

C. Network-aware Configuration

The previously described ADR algorithms adapt the transmission parameters of each node based on an estimated link budget. However, the performance of a LoRa network depends not only on such a budget but also on the occurrence of other simultaneous transmissions with the same spreading factor. This is determined by the spatial configuration of the deployed LoRa devices, namely, by the actual locations of both gateways and nodes. Indeed, it is possible to efficiently assign spreading factors to nodes when the location of all devices in the network is known [19]. Accordingly, we calculate the optimal distribution of spreading factors (Table II) that balances the collision probabilities between different spreading factors in the whole network. The probability of collisions in each spreading factor follows the unslotted ALOHA model [6, 13]. The optimal spreading factor distribution is obtained with a genetic algorithm, as in [19]. Once the distribution of spreading factors is known, our network-aware approach configures the parameters as follows. We first sort the nodes according to increasing distance[3] from the gateway. We then assign spreading factors to each node according to the optimal distribution: nodes closer to the gateway are assigned a lower spreading factor and those further away a higher spreading factor.

V. Performance Evaluation

We evaluate the performance of adaptive communications in LoRa networks through simulations. We start by describing the simulation setup, followed by the obtained results. We finally summarize the key findings from our experiments.

A. Simulation Setup

We employed FLoRa to evaluate the performance of LoRa networks with and without ADR. The number of LoRa nodes was varied from 100 to 700 in steps of 100 nodes. We used the European regional parameters for the LoRa physical layer detailed in Table III. Each LoRa node initially picked a random spreading factor and a transmission power level uniformly distributed within the permissible range, so as to cover all possible configurations available to the devices. The simulation scenario consisted of one LoRa gateway located at the center of the deployment area and connected to one network server, as illustrated in Figure 2. The Internet cloud component of the INET framework was used to model an ideal backhaul network with no packet loss and a transmission delay of 10 ms. The connections in the backhaul network were represented by Gigabit Ethernet links.

We considered a typical sensing application in our simulations. Specifically, each LoRa node sent a 20 byte packet after a time period drawn from an exponential distribution with a

[3] In practice, such a distance can be estimated based on the received SINR at the gateway.

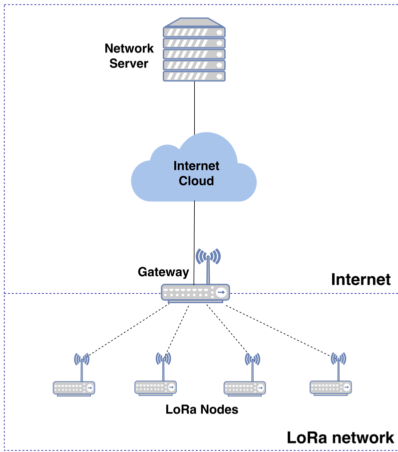


Fig. 2: Network topology.

mean of 1,000 s. The LoRa nodes and the gateway adhered to a 1% duty cycle restriction, as required for operation in the ISM bands [12]. We considered two different deployment scenarios, i.e., urban (U) and sub-urban (SU). The two scenarios differ in the path loss parameters used and, thus, in the size of the deployment area. According to the achievable communication range, the deployment area was set to 480 m by 480 m for the urban scenario and 9,800 m by 9,800 m for the sub-urban scenario. The size of the deployment area was selected as the maximum value that allows all nodes located within the square region to communicate with the gateway. Furthermore, we evaluated each scenario (urban and sub-urban) with different standard deviations in path loss (σ) set to represent: an ideal channel, a channel with moderate variability and a channel with typical variability (refer to Table IV for the details).

Nodes were placed at random locations uniformly distributed over the deployment area. Each individual experiment lasted for 12 days of simulated time. The simulations included a warm-up period of 2 days during which statistics were not collected while the network reached steady state. We run 30 iterations of each experiment according to the independent replication method. The plots report the average value obtained over all replications, along with the related 95% confidence intervals as error bars.

We evaluated the performance of LoRa networks with and without ADR. In networks without ADR, the mechanism was disabled at both the nodes and at the network server. When ADR was enabled, ADR-NODE ran on all nodes and the following variants of ADR at the network server were considered: (i) *ADR-NET*, as described in Section IV-A (i.e., Algorithm 2); (ii) *ADR+*, the variant of Algorithm 2 described in Section IV-B. Next, we compared the performance of networks configured with: (i) *ADR+*; (ii) *a network-aware approach* using the optimal distribution of spreading factors described in Section IV-C. In such a comparison, we considered an urban scenario (with variance in path loss σ set to 0 dB) and a dense deployment of LoRa nodes in a circle with radius of 50 m around a single gateway. We evaluated

TABLE III: Simulation parameters.

Parameter	Value
Carrier Frequency	868 MHz
Bandwidth	125 kHz
Code Rate	4/8
Spreading Factor	7 to 12
Transmission Power	2 dBm to 14 dBm

TABLE IV: Standard deviation of the path loss (σ) in dB for different deployment scenarios.

Scenario	Ideal	Moderate variability	Typical variability
Urban	0	1.785	3.57
Sub-urban	0	3.54	7.08

the network performance with different densities of devices by varying the number of nodes in the deployment area from 100 to 700.

Finally, we evaluated the following performance metrics:

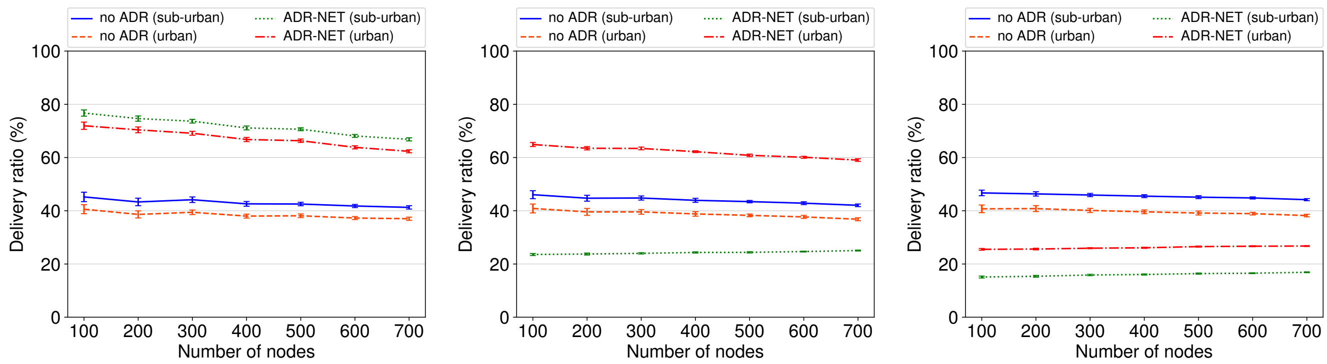
- *delivery ratio*, as the number of messages correctly received by the network server divided by the total number of messages sent by the end nodes;
- *energy consumption* per successful transmission, as the total energy used by all LoRa nodes divided by the total number of messages received by the network server.

B. Simulation Results

We start by evaluating the performance of networks with ADR and analyze the impact of channel variability. Next, we compare the performance of ADR+ with ADR-NET. Finally, we analyze the impact of network-aware configuration on the delivery ratio.

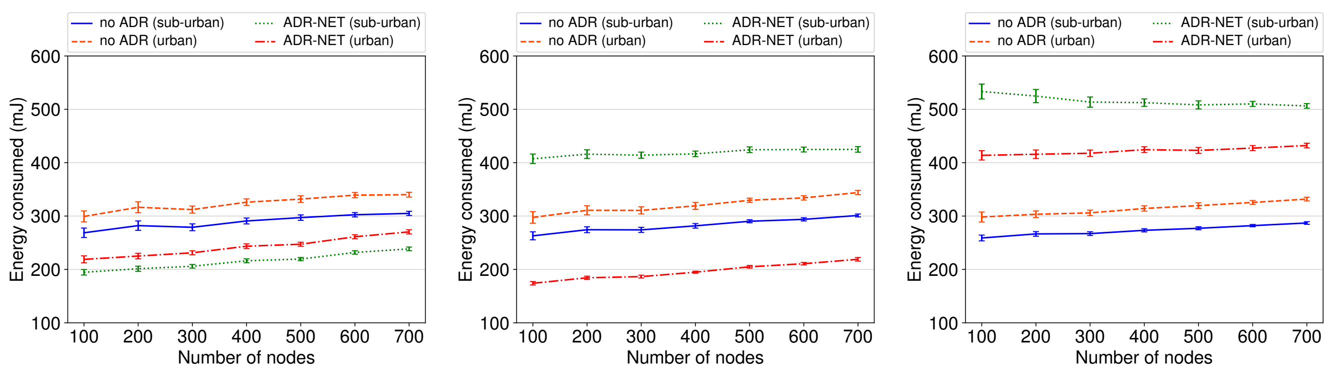
Impact of channel variability on ADR: We study the performance of ADR in urban and sub-urban scenarios. Figure 3 shows the impact of ADR on the delivery ratio as a function of the number of nodes for different channel conditions. We observe that the delivery ratio without ADR is around 40% for both urban and sub-urban scenarios. The delivery ratio slightly reduces as the number of nodes in the network increases. In the case of an ideal channel, ADR-NET performs better than a network without ADR. However, we observe that the delivery ratio with ADR-NET in sub-urban scenarios is worse than in a network without ADR when the channel has a moderate variability (Figure 3b). For the urban environment, ADR-NET still outperforms the case without ADR. Finally, Figure 3c clearly shows that networks with ADR perform worse than those without ADR for both scenarios. Thus, the delivery ratio obtained by ADR-NET decreases as the variance in the channel conditions increases.

We also evaluated the impact of ADR on energy efficiency. Figure 4 shows the related results, again, as a function of the number of nodes for different channel conditions. We clearly see how ADR results in a reduction of the energy consumption only in the networks with no channel variability. Instead, the energy usage of ADR-NET is higher than that achieved by using static parameters when channel conditions vary. This is mainly caused by the small delivery ratio achieved by ADR.



(a) (b) (c)

Fig. 3: Impact of ADR-NET on the delivery ratio in networks with different channel conditions: (a) ideal, (b) moderate variability, and (c) typical variability.



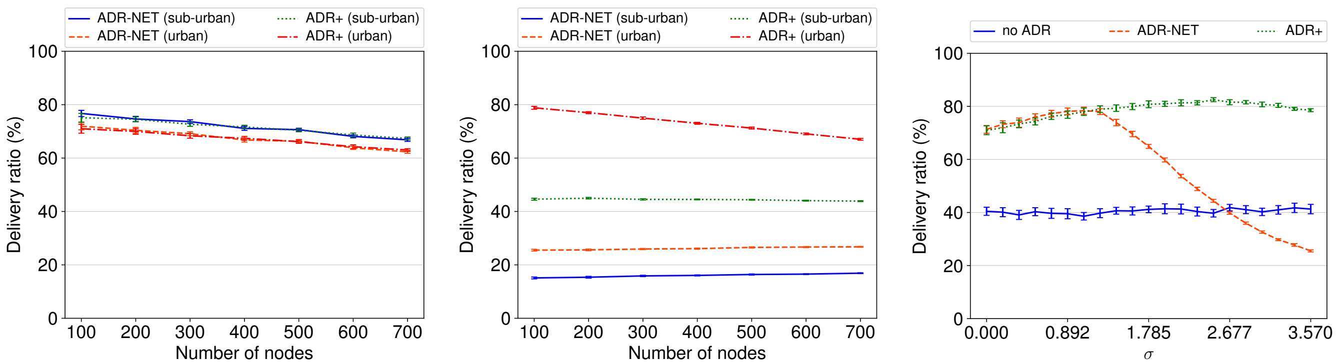
(a) (b) (c)

Fig. 4: Impact of ADR-NET on the energy consumption in networks with different channel conditions: (a) ideal, (b) moderate variability, and (c) typical variability.

Impact of different ADR algorithms: We now compare the performance of ADR-NET against our proposed ADR+. Figures 5a and 5b show the delivery ratio of the two schemes as a function of the number of nodes for two different channel conditions, i.e., ideal and typical variability respectively. The results obtained with ADR+ are comparable with those produced by ADR-NET with an ideal channel (Figure 5a). However, ADR+ achieves a significantly higher delivery ratio than ADR-NET in networks with non-zero path-loss ($\sigma > 0\,\text{dB}$). The improvement is at least 30% in all cases (Figure 5b). To evaluate the impact of channel variance on ADR performance, Figure 5c shows the delivery ratio (in a network of 100 nodes) as a function of σ in an urban scenario. We observe that ADR-NET and ADR+ perform similarly well when channel variability is low (i.e., until $\sigma < 1.25\,\text{dB}$) after which the performance of ADR-NET declines. ADR+ achieves a higher delivery ratio even when the channel variability is high.

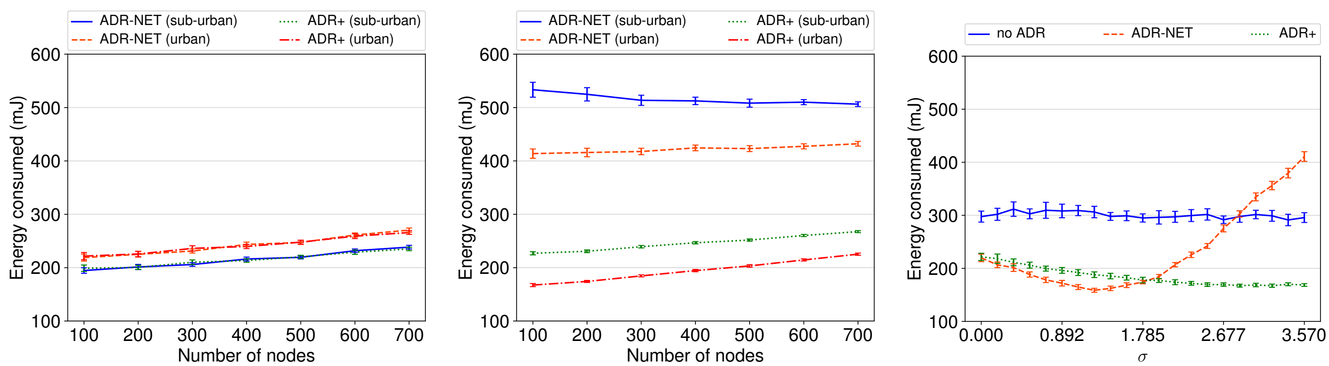
ADR+ also achieves better performance in terms of energy consumption, as shown in Figure 6. The results are similar to those obtained for the delivery ratio: the performance of ADR+ and ADR-NET is similar when the channel has no variability; however, ADR+ outperforms ADR-NET when the variability is different from zero. This is because the delivery ratio with ADR+ is much higher than that obtained by ADR-NET. Finally, Figure 6c shows the energy consumption in a network of 100 nodes as a function of σ in an urban scenario. The results clearly show how the performance without ADR is not affected by the variance in the path loss, while ADR+ and ADR-NET are sensitive to the actual value of σ. While ADR-NET achieves the best performance for $\sigma < 2.25\,\text{dB}$, its energy consumption sharply increases when the channel variability is high. In contrast, the energy consumption of ADR+ slightly decreases as σ increases. ADR+ lowers the energy consumption by using a more accurate estimation of the channel quality.

Impact of network-aware configuration: We compare the performance of the networks configured with ADR+ to that with optimal distribution of spreading factors. Figure 7 shows the delivery ratio as a function of the number of nodes. When the density of nodes is small (similar to [20]), the delivery ratio in both networks is comparable. However, as the number of nodes increases, the difference between the delivery ratios increases too – such a difference is about 20% in networks with 700 nodes. We observe that ADR+ tends



(a) (b) (c)

Fig. 5: Comparison of the delivery ratio of ADR-NET and ADR+ in networks with different channel conditions: (a) ideal, (b) typical variability. (c) Delivery ratio in urban scenarios (100 nodes) as a function of the path loss variance σ.



(a) (b) (c)

Fig. 6: Comparison of the energy consumption of ADR-NET and ADR+ in networks with different channel conditions: (a) ideal, (b) typical variability. (c) Energy consumption in urban scenarios (100 nodes) as a function of the path loss variance σ.

to assign the lowest spreading factor of 7 to all nodes close to the gateway. This allows nodes to communicate with the gateway with the lowest time on air. However, the collisions between the packets increase when the number of nodes with the same spreading factor increases substantially. Instead, the collision probability can be decreased by distributing the spreading factors among nodes. Consequently, the optimal distribution achieves a delivery ratio greater than 95% for all evaluated networks.

C. Discussion

We now discuss some of the key findings from the obtained results. We also highlight some solutions for the challenges in managing LoRa networks.

Low Transmission Power Trap: The ADR algorithm at a node, ADR-NODE, can increase the spreading factor of transmissions until the link budget allows the node to successfully transmit a frame to the network server. However, ADR-NODE can only increase the spreading factor and cannot change the transmission power. We observed that in certain cases a node can communicate with the gateway only if it transmits at a high transmission power and with a high spreading factor. In contrast, a node already transmitting with a low transmission power is unable to increase the transmission power of future messages. This creates what we call a "low transmission power trap", wherein a node can no longer communicate with the gateway as it is unable to increase its own transmission power. Our simulation results show that this condition occurs for about 36% of the nodes in the considered urban scenario and 30% of those in the sub-urban scenario.

Lower Performance of ADR in a Noisy Channel: ADR-NET performs well when there is no variance in the path loss, i.e., in an ideal channel with $\sigma = 0\,\text{dB}$. However, when the channel is noisy (with $\sigma > 0\,\text{dB}$), the original ADR-NET algorithm calculates SNR_{margin} based on the maximum SNR of received frames. The maximum SNR value tends to overestimate the channel quality. We find that the threshold to decrease transmission parameters in ADR-NET is satisfied when the $\max$ SNR value from last 20 samples is higher than 3 dB. Thus, ADR-NET decreases the transmission power and (or) the spreading factor too aggressively in networks with moderate to high path loss variance.

Network-aware Configuration vs Link Adaptation: ADR-NET and ADR+ improve the link budget by appropriately

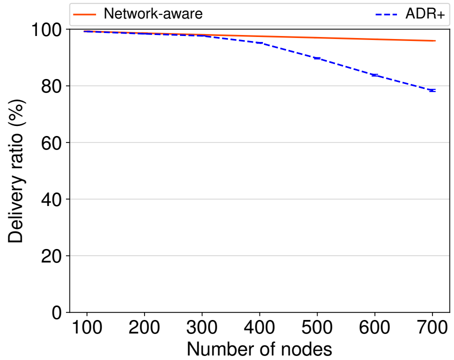


Fig. 7: Delivery ratio in networks using ADR+ and a network-aware approach for configuring the transmission parameters.

choosing the spreading factor and transmission power for each link. However, this approach does not take into consideration the possibility of collisions. As demonstrated by our experiments, the occurrence of collisions (due to many nodes using the same spreading factor) can significantly decrease delivery ratio in dense networks. Therefore, there is a need for an algorithm that configures transmission parameters based on the knowledge of the entire network. Clearly, such a knowledge of the network is not always available, for instance, as the actual locations of the nodes may be unknown. In such cases, ADR – in particular, ADR+ when the channel variability is high – still achieves satisfactory results for deployments that are not extremely dense.

VI. Related Work

Even though several articles studied the scalability of LoRa networks [2, 16, 21], none of them has considered the impact of ADR on performance. Bor et al. [6] propose an algorithm to select parameters such that transmission airtime and power are minimized. However, the authors themselves described the proposed algorithm as optimistic and impractical; their goal was to show that improvements in network capacity are possible with dynamic data rates. In contrast, we evaluate the ADR mechanism built into LoRaWAN and suggest a simple yet effective modification to improve its performance. Reynders et al. [19] presented and evaluated a mechanism to optimize the fairness of packet error rates among nodes with different spreading factors. We have applied a similar approach to derive the optimal distribution of spreading factors as a network-aware scheme in comparison with ADR+. However, the scenario considered in [19] is such that all nodes in the network can reach the gateway with every spreading factor and every power setting, i.e., all nodes are close to the gateway. In contrast, we consider a more realistic deployment scenario wherein nodes may only reach the gateway with specific spreading factors and transmission powers. Varsier et al. [20] analyzed the capacity limits of LoRaWAN networks for smart metering applications. The authors considered a distribution of spreading factors based on the median of the SNR values received at the gateway. However, the exact details on how to configure the transmission parameters were not provided in their work. In contrast, we have evaluated the impact of ADR on network performance and proposed modifications to the original ADR algorithms. Kim et al. [22] proposed a new ADR algorithm for LoRa networks at the nodes. Their algorithm requires an active feedback channel, i.e., an acknowledgment for every transmission. However, this mechanism would decrease the delivery ratio as downlink traffic has been demonstrated to have an impact on uplink throughput [10]. In contrast, we show that ADR improves the efficiency of LoRa networks without the need for acknowledgments.

There are a few articles which present simulation tools[4] to evaluate the performance of LoRa networks. Bor et al. [6] developed a Python-based discrete event simulator (called LoRaSim) to characterize the capacity of LoRa networks. However, the simulator supports only uplink transmissions from nodes to the gateway; thus, it cannot be used to evaluate ADR. Pop et al. [10] extended the LoRaSim simulator by adding support for downlink transmissions. The authors demonstrated that downlink transmissions in the network actually decrease the communication performance of the wireless connections. Van den Abeele et al. [8] presented a LoRa simulator based on ns-3. The work characterizes the scalability in scenarios with both uplink and downlink transmissions; however, it does not consider dynamic configuration of transmission parameters.

VII. Conclusion

In this article, we have evaluated the performance of LoRa networks using adaptive communications. In particular, we have considered the ADR mechanism that is built into LoRa networks. We have developed an end-to-end LoRa simulator and performed a thorough study of reliability and energy efficiency in LoRa networks. Our results showed that ADR is effective when the variance of the channel is zero or very low, while additional mechanisms are needed for highly-varying channels. We have proposed a modification to the link quality indicator, as well as a policy to increase transmission power at the nodes. However, a link-based adaptation is still not sufficient in dense networks. Thus, extending ADR with information on the collision probability and the distribution of parameters in the network allows to further improve performance in denser networks. In this regard, we plan on improving the ADR mechanism by investigating the optimal setting of the transmission parameters in a network. A possible option would be to balance the link budget for every link and the delivery ratio for the entire network. We consider this approach as a promising direction for future work.

Acknowledgments

This work was partially supported by the Academy of Finland grants number 299222 and number 305507. We would like to thank the CSC – IT Center for Science for provisioning the computational resources used for the evaluation.

[4]At the time of writing this article, only LoRaSim has been released as open-source software, while the source codes for the rest of the simulators are not publicly available.

REFERENCES

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, 2017.

[2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[3] LoRa Alliance, "LoRaWAN What is it? A technical overview of LoRa and LoRaWAN," Nov. 2015. [Online]. Available: https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRaWAN101.pdf

[4] LoRa Alliance, "LoRaWAN Specification (V1.0.2)," Jul. 2016.

[5] The Things Network, "The Thing Network Mission," https://github.com/TheThingsNetwork/Manifest/blob/master/Mission.md, 2015, [Online; accessed 29 May 2017].

[6] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '16. New York, NY, USA: ACM, 2016, pp. 59–67. [Online]. Available: http://doi.acm.org/10.1145/2988287.2989163

[7] D. Bankov, E. Khorov, and A. Lyakhov, "On the Limits of LoRaWAN Channel Access," in *Engineering and Telecommunication (EnT), 2016 International Conference on*. IEEE, 2016, pp. 10–14.

[8] F. V. d. Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *arXiv preprint arXiv:1705.05899*, 2017.

[9] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, and J. Melia, "Understanding the limits of LoRaWAN," *arXiv preprint arXiv:1607.08011*, 2016.

[10] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?" *arXiv preprint arXiv:1704.04174*, 2017.

[11] A. Varga, "OMNeT++," *Modeling and tools for network simulation*, pp. 35–59, 2010.

[12] LoRa Alliance, "LoRaWAN Regional Parameters," 2016.

[13] A. S. Tanenbaum, "Computer networks," Tech. Rep., 1996.

[14] "INET Framework," https://inet.omnetpp.org/.

[15] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall, 1996, vol. 2.

[16] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology," in *ITS Telecommunications (ITST), 2015 14th International Conference on*. IEEE, 2015, pp. 55–59.

[17] S. Corporation, "Sx1272/73 datasheet, rev 3.1, march 2017," http://www.semtech.com/images/datasheet/sx1272.pdf, 2017.

[18] The Things Network, "The Thing Network Wiki: Adaptive Data Rate," https://www.thethingsnetwork.org/wiki/LoRaWAN/ADR, 2017, [Online; accessed 29 May 2017].

[19] B. Reynders, W. Meert, and S. Pollin, "Power and Spreading Factor Control in Low Power Wide Area Networks," in *IEEE ICC 2017 SAC Symposium Internet of Things Track (ICC'17 SAC-7 IoT)*, 2017, pp. 1–5.

[20] N. Varsier and J. Schwoerer, "Capacity limits of LoRaWAN technology for smart metering applications," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

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

[22] D.-Y. Kim, S. Kim, H. Hassan, and J. H. Park, "Adaptive Data Rate Control in Low Power Wide Area Networks for Long Range IoT Services," *Journal of Computational Science*, 2017.