Simulation Modelling Practice and Theory 113 (2021) 102388

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Extending ADR mechanism for LoRa enabled mobile end-devices ☆

Vasileios Moysiadis [a], Thomas Lagkas [b], Vasileios Argyriou [c], Antonios Sarigiannidis [d], Ioannis D. Moscholios [e], Panagiotis Sarigiannidis [a,*]

[a] Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece
[b] Department of Computer Science, International Hellenic University, Kavala Campus, Greece
[c] Department of Networks and Digital Media, Kingston University, Kingston upon Thames, United Kingdom
[d] Sidroco Holdings Ltd, Nicosia, Cyprus
[e] Department of Informatics and Telecommunications, University of Peloponnese, Tripolis, Greece

ARTICLE INFO

Keywords:
LoRa
LoRaWAN
Adaptive data rate
Mobile devices
Energy consumption

ABSTRACT

A considerable percentage of Internet of Things end-devices are characterised by mobility, a feature that adds extra complexity to protocols used in Wireless Sensor Networks. LoRa is one of the newly introduced wireless sensor protocols, capable of delivering messages in long distances and consuming low energy, features that make it proper for low cost devices. Although LoRa was introduced as a technology for stationary devices, it can also be used for mobile devices of low speed. In this paper, we introduce an enhancement to Adaptive Data Rate (ADR) mechanism to enable mobile LoRa, by improving the connection reliability of mobile end-devices, while keeping energy consumption at low levels. Firstly, we propose the Linear Regression-ADR (LR-ADR) mechanism for the Network Server side to smooth the Signal to Noise Ratio (SNR) estimates per gateway and predict the SNR of the next transmission. Secondly, we propose the Linear Regression + ADR (LR+ADR) mechanism, an adaptive method for the end-device side to regain the connectivity faster with the Network Server. We conducted simulation modelling to evaluate the performance of our implementation while we compared our results with four alternative solutions ADR, ADR+, EMA-ADR, G-ADR. The results prove that our first approach (LR-ADR) performs better than the best competitor, and our second approach (LR+ADR) brings an additional improvement in terms of Packet Delivery Ratio (PDR), while they retain the Energy Consumption per Packet Delivered (ECPD) at low levels. In particular, in a scenario that mimics real world conditions, LR+ADR presents an increase of up to 520% for PDR compared to the original ADR and an improvement of up to 38% compared to the best competitor (G-ADR). Moreover, it reduces ECPD up to 74% compared to the original ADR, while keeping it at the same level with the best competitor (G-ADR).

1. Introduction

The Internet of Things (IoT) promises to change the world by participating in every part of our society. Smart Farming, Industry 4.0, Smart Cities, Smart Metering are some of the numerous examples of the expected impact of IoT in the upcoming years. A massive number of devices will be involved in the implementation of this vision. In addition, a considerable percentage of end-devices is

☆ This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957406 (TERMINET).

* Corresponding author.
E-mail addresses: bmoisiadis@uowm.gr (V. Moysiadis), tlagkas@cs.ihu.gr (T. Lagkas), vasileios.argyriou@kingston.ac.uk (V. Argyriou), asarigia@sidroco.com (A. Sarigiannidis), idm@uop.gr (I.D. Moscholios), psarigiannidis@uowm.gr (P. Sarigiannidis).

https://doi.org/10.1016/j.simpat.2021.102388
Received 31 May 2021; Received in revised form 30 July 2021; Accepted 1 August 2021
Available online 8 August 2021
1569-190X/

characterised by mobility, an extra parameter that makes the protocols used for wireless connectivity more complex. Moreover, most of them are expected to work in the field, without or with minimal human intervention, working with batteries daily for a few years, taking and transmitting measurements in long distances every few minutes or once a day. To fulfil these requirements, we need to reduce their energy consumption not only by improving their hardware specifications but also by improving their software. The energy consumption for a device to send data through a wireless network is quite important compared to overall energy consumption. Recently, many new wireless network technologies, namely Long Range (LoRa), Narrow Band IoT (NB-IoT), SigFox, are trying to solve this problem by offering wireless communication in long distances with low energy consumption.

This paper focuses on Long Range (LoRa) and investigates its capabilities to support mobile end-devices with energy restrictions. LoRa is a Low Power Wireless Area Network (LPWAN) designed to consume low energy on end-devices and achieve transmissions over long distances with low data rates. Since it uses an open spectral band without license fees, it is an excellent candidate for multiple applications deployed by any individual with minimal cost. It is recognised globally as an open wireless network that is already used on various applications in multiple domains. Although LoRa was not initially designed for mobile devices, it is used in various applications with needs in mobility. Such applications can be mechanical equipment tracking, measurements from mobile sensors on smart city scenarios, or personnel tracking in case of emergency disasters. For example, the authors in [1] propose a novel dissemination protocol to exchange information through LoRa multihop transmission in case of emergency scenarios. The authors in [2] present the advantages of a mobile gateway instead of a static one in cases such as smart live monitoring system of large areas. The authors in [3] present a high precision and energy efficient localisation system for mobile IoT devices based on Global Navigation Satellite System (GNSS) combined with a LoRa module. In [4] the authors present a LoRa application for animal tracking, tested in a real environment with wearable devices for cows. The authors, in [5] present a LoRa based air quality monitoring system that uses an Unmanned Aerial Vehicle (UAV) to accumulate measurements by flying over the area. Finally, in [6] the authors proposed a system based on LoRa for the wireless communication between vehicles and stationary gateways as an alternative solution to other wireless protocols like LTE.

Furthermore, energy consumption is one of the significant parameters that concerns many researchers nowadays in order to reduce the impact of humanity on Greenhouse emissions. From electric vehicles to electrical house equipment and the numerous end-devices of the Internet of Things, recent trends look forward to reducing the energy they consume. Moreover, this will lower the economic cost for the owner, as an energy-efficient device can work more with less energy. This feature is significant, especially for small end-devices of the Internet of Things regarding the constraints they have in energy consumption, as many of them work on batteries and expected to operate in the field for a few years without charging or human intervention. Except for the hardware features that specify the energy consumption of a specific device, wireless protocols also impact significantly on the total energy consumption.

LoRa is one of the most promising wireless technology for the Internet of Things. The LoRaWAN protocol lies on the MAC layer and is responsible for the communication between end-devices and the gateways as well as the Network Server (NS). The primary issue in end-devices using LoRa is the energy constraints since most of these devices are working on batteries. That makes it crucial to design new algorithms taking in mind to keep energy consumption at low levels. The Adaptive Data Rate (ADR) mechanism, which is part of the LoRaWAN protocol, tries to select the best parameters combination to achieve collision avoidance, increase network capacity, and reduce the energy consumption of end-devices. Although ADR is not designed for mobile end-devices [7], it would be a useful ability if we could extend it to handle them more efficiently. As there is not much research effort in the field of LoRa enabled mobile end-devices, more needs to be done in this direction.

Our approach is aiming to support applications for mobile devices that move at low speed. Potential applications could be the transmission of accumulated measurements from vehicles or moving equipment with low movement speed. Such applications will be expected to be common in Smart Farming scenarios where local agricultural associations may need to take environmental measurements or information for their vehicles or mechanical moving equipment. Such a scenario can be deployed in large agricultural areas which are typically almost flat, so they can be covered with a few LoRa gateways since they do not have significant obstacles. Thus, the LoRa mobile end-devices could always be in almost line of sight with at least one gateway. Although LoRa is not designed for mobile devices, it can efficiently work on mobile devices with low speed. In addition, it would be useful if we adjust the mechanism of data transmission in order to enhance its reliability, efficiency and decrease the energy consumption of end-devices if possible.

In this paper, we are introducing an enhancement of the ADR mechanism in order to support mobile end-devices more efficiently, and it is twofold. Firstly, we propose the Linear Regression-Adaptive Data Rate (LR-ADR) mechanism, which introduces a modification of the original ADR on the Network Server side. In LR-ADR, the Network Server is responsible for selecting the optimal parameters for Spreading Factor (SF) and Transmission Power (TP), while the end-devices are running the default ADR mechanism. Secondly, we propose the mechanism Linear Regression + Adaptive Data Rate (LR+ADR), where we introduce modifications on the end-device side, taking in mind the constraints of LoRa end-devices, while keeping the same changes on the Network Server side as LR-ADR does. Both of our approaches focus on increasing Packet Delivery Ratio (PDR) and are shown through simulation modelling to reduce mobile LoRa enabled end-devices' energy consumption by selecting the appropriate parameters for each one depending on the Signal to Noise Ratio (SNR) values of the packets captured by the nearby gateways.

The contributions of our work are summarised in the following:

- We are introducing two alternative extensions for ADR mechanism of LoRaWAN, adaptable to mobile end-devices.
- Both of the proposed mechanisms are shown through comparative simulations to be energy efficient, in terms of the total energy consumed in the network, while increasing the Packet Delivery Ratio (PDR).

- The proposed LR-ADR mechanism has backward compatibility with end-devices running the default ADR mechanism.
- The proposed LR+ADR mechanism from the end-device side has backward compatibility with Network Servers running the default ADR mechanism.

The rest of this paper is organised as follows: In Section 2 the most relevant work for LoRa enabled mobile end-devices is presented. In Section 3, we discuss the most important features used of LoRa. A thorough analysis of the original ADR mechanism is also presented. Section 4 analyses in depth the proposed LR-ADR and LR+ADR mechanisms. In Section 5, we provide the main parameters of the simulation modelling. Next, in Section 6, we present the evaluation between our proposed ADR mechanisms and other four well-known alternatives, ADR, ADR+, G-ADR, and EMA-ADR. Finally, Section 7 concludes this paper.

2. Related work

In this section we are presenting the most relevant work for LoRa enabled mobile end-devices. Firstly, we focus on existing research efforts for LoRa networks with mobile end-devices or mobile gateways. Only one of them deals with the enhancement of the ADR mechanism. In addition, we discuss some interesting research efforts that deal with ADR mechanism enhancement, but they do not refer to mobile end-devices.

In [8] the authors proposed two alternative ADR mechanisms, namely Gaussian filter-based ADR (G-ADR) and Exponential Moving Average-based ADR (EMA-ADR). Both of them reduce the convergence period for SF and TP of end-devices compared with original ADR and ADR+. G-ADR and EMA-ADR make changes on SF and TP only when an ADRACKReq MAC command with the ACK bit enabled is received from an end-device. In addition, both of the mechanisms offer a better packet success ratio while reducing energy consumption. The evaluation was performed in a static scenario and a mobility scenario. In more detail, G-ADR suggests modification on the ADR mechanism on the Network Server side. At the first step, it calculates the mean (μ) and the variance (σ) of the last $M = 20$ SNR values and then computes the average SNR value by using only those that are within the effective range of $\mu - \sigma$ and $\mu + \sigma$. The rest of the mechanism is the same as the original ADR. EMA-ADR uses an exponential moving average smoothing function in order to smooth the SNR signal of the last $M = 20$ values. Both G-ADR and EMA-ADR, adjusts SF and TP only when an ADRACKReq MAC command with the ACK bit enabled is received from an end-device.

The authors in [7] present the Enhanced ADR (E-ADR) mechanism, which tries to minimise transmission time, energy consumption, and the packet loss of mobile end-devices. The concept of their work is to select the optimal SF and TP based on the location of static or mobile end-devices. For that reason, they used a trilateration method based on RSSI values to predict the position. At the next step, the mechanism has to decide whether or not the end-device has moved to a new zone, so it has to change its configuration. The authors tested the proposed mechanism in three use cases: a cleaning robot use case, a drone inspecting the parcels, and a vegetable and fruit maturity monitoring robot.

In [9] the authors present an energy-efficient mechanism for LoRa networks that provides a system for locating and rescuing people. The localisation method based on trilateration and Time Difference of Arrival (TDoA) of the sent packets. The proposed method was tested with various wearable end-devices and proves that it can reduce energy consumption compared with the traditional localisation method with GPS.

The authors in [10] propose LoRaUAV, based on a WiFi ad hoc network with Unmanned Aerial Vehicles (UAVs) as the gateways, to support mobile end-devices. The presented algorithm focuses on changing periodically the topology of the gateways to adapt to the movements of end-devices. The algorithm is fully distributed, and the evaluation gives more reliable performance metrics for the Average End-to-End Packet Reception Ratio (AE-PRR) and the Average Total Delay (ATD).

To overcome the connectivity problems due to interference from obstacles or deep fading, the authors in [11] present a new data forwarding scheme. In this vein, mobile end-devices can transmit packets to nearby end-devices instead of storing data until they have contact with a gateway. To achieve that, the authors propose two new LoRaWAN classes, the Modified Class-C and the Queue-based Class-A. A simulation experiment was set up, based on the London bus network. The results show a significant reduction in data transmission delays and an improvement in the throughput of data transfer.

The authors in [12] propose an enhanced ADR mechanism based on the adaptation of Coding Rate (CR) and Transmission Power (TP), trying to optimise the trade-off between the delivery ratio and the energy consumption. Due to the capture effect phenomenon, when two colliding signals have almost the same strength, they are both lost. Thus, considering this, the proposed algorithm tries to leverage the values for Spreading Factor, Transmission Power and Coding Rate for the uplink signals from all end-devices, resulting in an increment of the delivery ratio. The authors prove the effectiveness of their algorithm by deploying an empirical study. In addition, they provide large-scale simulations showing that their method outperforms the existing schemes. Their approach is tested only on stationary end-devices with one gateway.

An adaptive data rate control mechanism is used in [13] to avoid congestion in a LoRaWAN network. Firstly, the authors propose a congestion classifier to estimate network congestion based on logistic regression. Then, the data rate is controlled according to the congestion status, and a backoff mechanism is used to avoid congestion on the next uplink message. All of the computation efforts for the congestion classifier occurs in the Network Server. Thus, the end-devices does not consume additional energy. The authors prove the effectiveness of the proposed mechanism by providing a performance analysis.

In [14], the authors present an enhancement of the ADR mechanism that reduces the convergence period, so the end-devices can reach the optimal data rate earlier. For the end-device part of the mechanism, they propose an alternative method where the network server is responsible for sending a LinkADRReq message to reduce the data rate in the next transmission. For that reason, the network server tracks the packet delivery ratio (PDR), and a threshold of 80% is used as a trigger. On the network server part

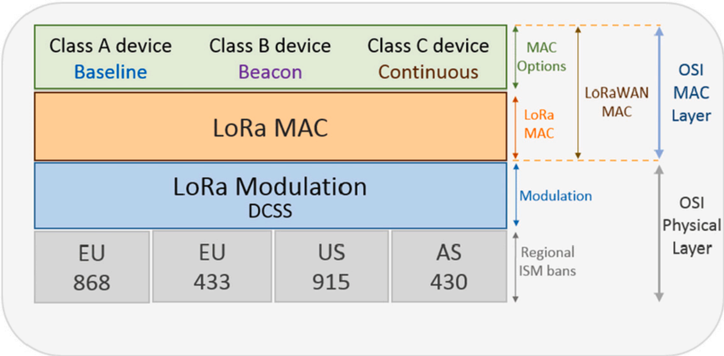


Fig. 1. LoRa and LoRaWAN protocol stack [22].

of the mechanism, the ADR mechanism first waits for at least five uplink frames from the last data rate change of an end-device. If the standard deviation of those last SNR values is less than 2.5 dB and if those values differ from the values before the data rate change, the ADR mechanism is triggered to estimate the optimal data rate.

Finally, the authors in [15] propose an enhancement of the original ADR mechanism aiming to increase reliability and energy efficiency. The proposed ADR+ mechanism slightly changes the part of the original ADR on the Network Server side. Instead of using the maximum value of the last 20 SNR signals, the Network Server calculates the average value and uses it to estimate the optimal SF and TP for the specific end-device. The authors evaluate the effectiveness of the new mechanism through simulations. For this purpose, they develop the FLoRa framework, an open-source event-driven simulator based on OMNET++.

It should be noted that only the first five of the above research efforts deals with mobile end-devices, and only one of them deals with increasing reliability and reducing energy consumption on mobile end-devices by adapting ADR. However, it is only tested at low speeds with a small number of gateways. Thus, more research effort is required in this direction. Our approach focuses on mobile end-devices, and it differentiates from similar efforts since it does not calculate the optimal values for the SF and TP based on the current uplink message, but it is trying to predict the next SNR value. Thus, it is able to calculate the optimal values for SF and TP for the next expected uplink message in order to keep connectivity in the rapidly changing network topology. In addition, most of the recent research efforts use simulation areas with only one gateway. In this paper, we are using multiple gateways in an area where mobile end-devices move from one gateway to another, making the environment more complex. We are testing and evaluating through simulation modelling the performance at various speeds and packet transmission intervals. Furthermore, our proposed schemes are compatible with all existing LoRa enabled mobile end-devices by altering the existing ADR mechanism without introducing new routing protocols.

3. LoRa specifications — Background

LoRa belongs in the category of Low Power Wireless Area Networks (LPWANs) and is capable of transmitting messages in long distances with low energy consumption sacrificing high data rates. LoRa stands for Long Range, and was firstly introduced by Cycleo, before SEMTECH acquired copyrights. It can reach a transmission distance of up to 15 km in rural areas with low energy consumption [16]. Thus, it is suitable for various applications on which end-devices can operate in the field for a few years without or with minimal human intervention. For example, Smart Farming [4,17], Marine Aquaculture [18], Smart Cities [19,20], Smart Grid [21] are some of the domains suitable for LoRa technology.

It uses a free license band depending on the region it operates. In Europe it operates under the free license band of 868 MHz while in US uses 915 MHz and in Asia the 430 MHz (Fig. 1). At the physical layer, the transmission is based on Chirp Spread Spectrum (CSS) modulation. Moreover, it uses various parameters for optimal configuration which are the Spreading Factor (SF), the Transmission Power (TP), the Coding Rate (CR), and the Bandwidth (BW).

In more detail, the LoRa specifications of the physical layer support six different spreading factors from SF7 to SF12, while the newest chipset SX126x series from SEMTECH supports seven different Spreading Factors from SF6 to SF12. Lower Spreading Factors mean less time on air, higher data rate, and low energy consumption, but lag behind long distance transmission. Higher Spreading Factors means more time on air, lower data rate, high energy consumption, but the transmitted signal can reach long distances. Eq. (1) calculates the time on air for a symbol transmitted for different spreading factors.

$$T_s = \frac{2^{SF}}{BW} \tag{1}$$

Where T_s is the time on air for one symbol to transmit over a channel with bandwidth BW, and with a spreading factor SF.

The parameter Transmission Power (TP) refers to the energy the end-device puts into the transmitted signal. LoRa specifications define that TP can take values from −4 dBm to 20 dBm with steps of 1 dBm, but due to hardware limitations, the effective values are from 2 dBm to 17 dBm [23]. In Europe, the higher allowed value for TP is 14 dBm. The higher the TP, the strongest the signal

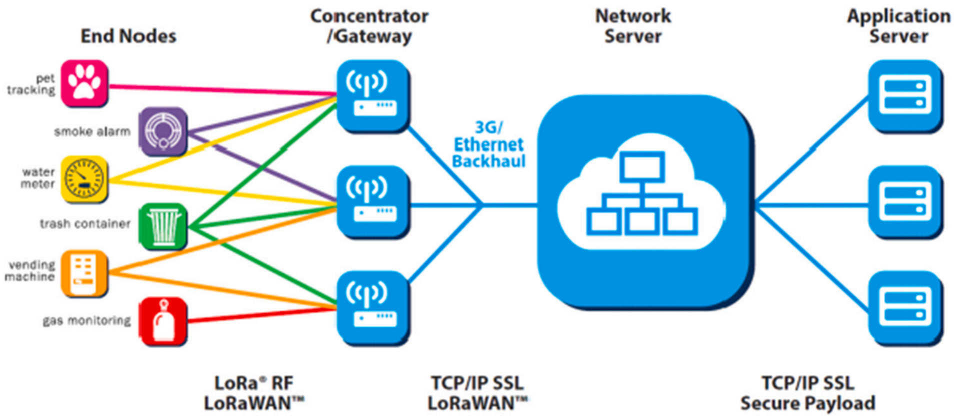


Fig. 2. LoRaWAN Architecture.

sent, which in turn results in higher transmission distance. When the TP is low, the signal uses less energy, but it cannot reach long distances.

Higher BW results in a higher data rate which gives lower time on air for the same SF. Instead, lower BW gives a lower data rate and results in higher transmission time for the same SF. LoRa specifications define three different bandwidths, 125 kHz, 250 kHz, and 500 kHz.

Coding Rate (CR) is used for error detection and correction in case of a burst of interference. LoRa uses Hamming Code as the forward error correction technique [24]. The default value is 4/5, while it can take different values (4/5, 4/6, 4/7, 4/8). More specifically, it adds additional symbols in the transmitted signal, which offers error detection and correction, but it also increases time on air.

On the MAC layer, the LoRaWAN protocol is responsible for the communication between end-devices and the gateways. In addition, LoRaWAN defines three types of devices with different capabilities. Class A devices correspond to end-devices with limited capabilities, and they can mainly send messages and receive a short downlink message only after a transmission. Class B devices are synchronised with the network sending beacons periodically. They can transmit messages and receive downlink messages at any time but only on specific time slots. Lastly, Class C devices are the most energy consuming, as they are able to receive or transmit messages at any time. For example, the LoRa gateways belong in Class C.

LoRaWAN architecture is a network topology star of stars, where one or more LoRa gateways relay messages between end-devices and the Network Server. Gateways are connected through IP protocol with the Network Server and act as a bridge for RF packets of LoRa to IP packets. A simple LoRaWAN architecture is shown in Fig. 2, where various devices are able to transmit and receive messages through one or more nearby gateways. The gateways are capable of demodulating the received messages and uploading them to a Network Server. The Network Server is responsible for resolving the deduplication of the same messages that come from the same end-device but are uploaded from a different gateway. Finally, it sends the message to the Application Server to which the end-device belongs.

The Adaptive Data Rate (ADR) mechanism is part of the LoRaWAN protocol and focuses on increasing the energy efficiency of Class A end-devices. It is divided into two parts, one that runs on the Network Server (NS) and one that runs in the end-devices (ED). Since energy constraints characterise most Class A EDs, the ADR mechanism on their side has been kept simple to avoid additional computational resources, which result in more energy consumption. In [25] the authors present an extensible analysis of ADR and discuss it regarding LoRaWAN Specifications (v1.1).

The ADR algorithm from the Network Server side is summarised as follows. When a new packet from the queue is processed, the NS finds the gateway that receives the signal with the maximum Signal to Noise Ratio (SNR) value. This gateway will be used to transmit back the acknowledgement message with the ACK bit enabled. In addition, the SNR value is added in a queue with the last known SNR values for the specific ED. The Network Server checks this queue periodically every $M = 20$ received packets from a specific ED and calculates the SNR_{margin} based on Eq. (2).

$$SNR_{margin} = SNR_{max} - SNR_{req} - margin_{dBm} \tag{2}$$

Where SNR_{max} is the maximum SNR found from the last $M = 20$ signals, SNR_{req} is the SNR value of the requested signal shown in Table 1. The parameter $margin_{dBm}$ is a specified value threshold with the default value of 10 dBm for the original ADR mechanism. The calculated SNR from Eq. (2) is used to calculate the value of N_{step} based on Eq. (3).

$$N_{step} = round\left(\frac{SNR_{margin}}{3}\right) \tag{3}$$

N_{step} is used to increase or decrease SF and TP of the ED and select the optimal ones in order to reduce energy consumption and increase network capacity. In more detail, when N_{step} is positive, the NS first reduces the SF, so many times as N_{step} indicates.

Table 1
Required SNR for effective signal demodulation on corresponding SF.

SF	7	8	9	10	11	12
SNR_{req} (dB)	−7.5	−10	−12.5	−15	−17.5	−20

If SF reaches the lower value (SF7), the NS reduces the TP by steps of 3 dBm unless it is already at the minimum level (2 dBm). Instead, when N_{step} is negative, the NS first increases the TP by steps of 3 dBm so many times as N_{step} indicates. If TP reaches the maximum value (14 dBm), the NS increases the SF unless it is already at the maximum level (SF12). Finally, a TX_CONFIG command is transmitted to the end-device and the new parameters of SF and TP used for the next uplink message.

The ADR algorithm on the end-device is implemented to be as simple as possible to consume less energy. It only tries to increase TP and SF when the end-device loses connectivity with the Network Server. To achieve that, it uses three parameters ADR_ACK_LIMIT, ADR_ACK_DELAY, and ADR_ACK_CNT. The ADR algorithm increases the parameter ADR_ACK_CNT by one every time the end-device transmits a packet without receiving an acknowledgement message with the ACK bit enabled from the NS. Besides, when it receives an acknowledgement message, ADR_ACK_CNT resets to zero. The parameter ADR_ACK_CNT then is used in comparison with ADR_ACK_LIMIT and ADR_ACK_DELAY. Firstly, if ADR_ACK_CNT become equal to ADR_ACK_LIMIT, it signals that there is a connectivity problem. At this point, it transmits the next packet with ADRACKReq bit enabled. Until no acknowledgement message received from the NS, the ED continuous to send the following ADR_ACK_DELAY packets with ADRACKReq bit enabled before it starts to increase SF and TP. When ADR_ACK_CNT become equal to ADR_ACK_LIMIT + ADR_ACK_DELAY, it first increases by three the TP and waits for another ADR_ACK_DELAY packets unless the connectivity is resolved. This procedure continues until TP reaches the maximum level, 14 dBm for Europe. At this point, the ED increases by one the SF until it reaches the maximum level (SF12) or the connectivity is resolved. If both SF and TP reach the maximum values, the ED does not take any additional action. The default values for ADR_ACK_LIMIT and ADR_ACK_DELAY for the ADR algorithm are 64 and 32, respectively.

4. Proposed ADR mechanisms

Our work deals with the enhancement of ADR mechanism in order to keep a reliable connection between a mobile end-device and the Network Server. Our approach aims to increase the packet delivery ratio and retain low the energy consumption ratio per successfully delivered packet. As mentioned in the previous section, ADR is responsible for selecting SF and TP parameters such as the end-device can send packets efficiently for the reception from a nearby gateway. Under this perspective, we are proposing two schemes, namely, Linear Regression-ADR (LR-ADR) and Linear Regression + ADR (LR+ADR), which both enhance the original ADR algorithm. In LR-ADR, we are proposing modification only on the Network Server side of the ADR algorithm, while in LR+ADR, we are proposing additional modifications on the end-device side. In this section, we are analysing in detail both of our proposed schemes.

4.1. LR-ADR mechanism

The first of our proposed schemes, LR-ADR, uses simple linear regression to smooth the SNR signal between an end-device and every separate gateway of the network. In [26] linear regression is proposed as a local indoor positioning method based on RSSI measurements. In our case, we are using simple linear regression locally for the last ten SNR signals in each gateway. Although SNR and RSSI are not linear compared with the distance from a gateway, they may be considered locally linear. Although SNR and RSSI values have a lot of noise, they tend to be linear between close positions of a mobile end-device, especially when the end-device is not too close to a gateway. Based on the smoothing results of linear regression for the last transmitted packet, the Network Server selects the gateway with the best SNR to send the ACK message and any required adjustments to SF and TP. In addition, we calculate the average of the next estimated SNR value from the gateways received the last packet and add it to a queue which will be used in the next step to calculate the SNR_m.

Our proposed LR-ADR algorithm is as follows :

1. A packet sent from end-devices may be delivered to the Network Server from one or more gateways. Since we are dealing with mobile end-devices, the SNR may vary in time for any different gateway. Thus, we are applying simple linear regression to smooth the SNR signal from the last packets received for all different gateways. We apply simple linear regression to the last 10 SNR values for each gateway per end-device. Thus, for each gateway that received the last packet, we have to find the parameters α and β from Eq. (4).

$$SNR_T = \beta T + \alpha \tag{4}$$

Where SNR_T is the SNR value at time T. To estimate the values of α and β we are using Eqs. (5) and (6).

$$\alpha = \overline{SNR} - (\beta \overline{T}) \tag{5}$$

$$\beta = \frac{\sum_{i=1}^{n}(T_i - \overline{T})(SNR_i - \overline{SNR})}{\sum_{i=1}^{n}(T_i - \overline{T})^2} \tag{6}$$

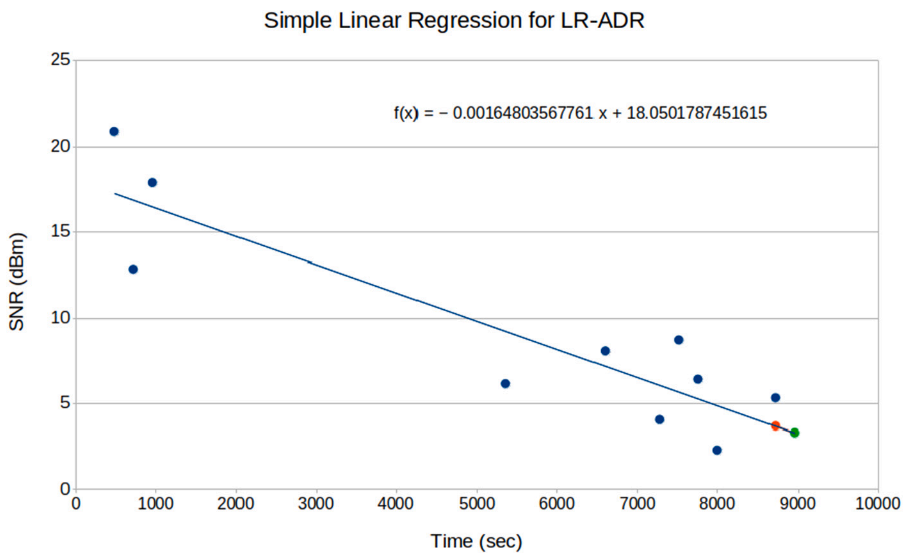


Fig. 3. Simple Linear Regression for SNR values of a Gateway.

Where T_i and SNR_i, is a specific time and the corresponding SNR at that time. The $\overline{T}$ and $\overline{SNR}$ is the average of T_i and SNR_i respectively.

After the estimation of α and β, NS calculates the smoothed SNR value for the current time based on Eq. (4). The Network Server repeats this smoothing procedure for the last packet delivered from different gateways. After all smoothing estimation for SNR, NS finds the maximum value from them and uses the corresponding gateway as the best candidate to transmit the downlink message with the ACK bit enabled to the end-device.

2. Since most of the LoRa devices transmit periodically with a fixed time interval, the Network Server first tracks the transmission period T_{period} for the device by calculating the minimum time interval during the last ten transmissions. Besides, to estimate the next values for SF and TP, we are calculating the next expected SNR value for each gateway that transmitted the last packets from the end-devices. For this reason, we are using Eq. (7).

$$SNR_N = \beta(T + T_{period}) + \alpha \tag{7}$$

As mentioned above, in mobile end-devices scenarios, they can easily lose connectivity with a gateway due to their movement or by an obstacle that may be found between it and the gateway. Considering that we want to keep the connection between the end-device with at least one gateway we are not using the maximum SNR_N value calculated in the previous step, since this would be an optimistic decision. Instead, the average value of SNR_N is calculated from all the gateways received the last packet and is added to a list for use in the next step.

3. The next part of the LR-ADR algorithm is the same as the original ADR. The NS calculates the average from the list with the SNR values from the last $M = 20$ packets. Finally, NS calculates SNR_{margin} value with Eq. (2), and then Eq. (3) is used to select the appropriate SF and TP for the end-device to prepare it for the following transmission.

An example of how simple linear regression is used in the LR-ADR mechanism is shown in Fig. 3. The x-axis represents the time, while the y-axis represents the SNR values. As described above, the last ten SNR values (blue dots) are used to calculate the simple linear regression function shown in the figure. The line shown in the figure has the minimum distance from all these SNR values. In essence, simple linear regression is used to smooth the last SNR value and to predict the next expected SNR value from the specific end-device. In our example, the smoothed current SNR value (red dot) is calculated with that function, so it lies on that line. In addition, the same function calculates the next expected SNR value (green dot) on the time after the calculated minimum time interval of the last received packets.

4.2. LR+ADR mechanism

To enhance the efficiency of LR-ADR we are suggesting a simple modification from the end-device side of the original ADR algorithm. The proposed changes aim to reduce the convergence period in case of an end-device loss connection with the Network Server. The default values for parameters ADR_ACK_LIMIT, ADR_ACK_DELAY are 64 and 32 respectively, resulting in a long time period for an end-device to regain connectivity by increasing step by step SF and TP. As mentioned in [25], by decreasing the values of ADR_ACK_LIMIT, ADR_ACK_DELAY, the convergence period is also reduced without an impact on energy consumption. Especially in mobile end-devices where the SNR values are changing more often due to movement or obstacles, a more efficient method to

Table 2
Possible combinations of the parameters ADR_ACK_LIMIT, ADR_ACK_DELAY.

ADR_ACK_LIMIT	16	32	64
ADR_ACK_DELAY	8	16	32

regain connectivity is essential. In addition, LoRa specifications v.1.1 specify that the parameters ADR_ACK_LIMIT, ADR_ACK_DELAY can be changed with the ADRParamSetupReq command from the NS. According to this command, they both can take values as a power of two as shown in Eqs. (8) and (9).

$$ADR\_ACK\_LIMIT = 2^{Limit\_exp} \tag{8}$$

$$ADR\_ACK\_DELAY = 2^{Delay\_exp} \tag{9}$$

Where Limit_exp and Delay_exp can take values from 0 and 15, corresponding to a range of 1 to 32768 for the ADR_ACK_LIMIT and ADR_ACK_DELAY parameters.

Our approach does not use the command ADRParamSetupReq from the Network Server, but proposes an adapting method to change these parameters if frequently lost packets are detected or not from the end-device side. Thus, in the proposed LR+ADR mechanism, the parameters ADR_ACK_LIMIT, ADR_ACK_DELAY can take a set of values as shown in Table 2.

In more detail, we are introducing a new parameter, ADR_ACK_RECEIVE_CNT, responsible for counting the concurrently last received acknowledgement packets. ADR_ACK_RECEIVE_CNT is increased by one if the Network Server successfully receives the last sent packet and an ACK command is returned. Otherwise, ADR_ACK_RECEIVE_CNT is set to zero. When the parameter ADR_ACK_RECEIVE_CNT becomes greater than ADR_ACK_DELAY, it signals that we have a reliable connection and the parameters ADR_ACK_DELAY and ADR_ACK_LIMIT are increased if they do not already have the maximum values. To decrease the same parameters, we are using the existing ADR_ACK_CNT parameter as a counter. More specifically, when ADR_ACK_CNT becomes greater than ADR_ACK_DELAY+ADR_ACK_LIMIT they are both divided by two for the ADR mechanism to become more elastic next time it loses connectivity. Based on this algorithm, when dealing with mobile end-devices that face rapid changes to their SNR values due to their position changing, they can sooner regain the connectivity with the Network Server as it increases SF and TP without waiting a long time.

The LR+ADR algorithm from the end-device side is described in the following algorithm.

Algorithm 1: LR+ADR mechanism from the end-device side

Sending Packet

```
ADR_ACK_CNT++;
if (ADR_ACK_CNT == ADR_ACK_LIMIT) then
    sendNextPacketWithADRACKReq = true;
end
if (ADR_ACK_CNT >= ADR_ACK_LIMIT + ADR_ACK_DELAY) then
    ADR_ACK_CNT = 0;
    if (ADR_ACK_LIMIT > 16) then
        ADR_ACK_DELAY = ADR_ACK_DELAY / 2;
        ADR_ACK_LIMIT = ADR_ACK_LIMIT / 2;
    end
    Increase TP or SF if possible
end
```

Receiving Acknowledgement

```
if (ADR_ACK_CNT == 1) then
    ADR_ACK_RECEIVE_CNT++;
else
    ADR_ACK_RECEIVE_CNT = 0;
end
if (ADR_ACK_RECEIVE_CNT > ADR_ACK_DELAY) then
    ADR_ACK_RECEIVE_CNT = 0;
    if (ADR_ACK_LIMIT < 64) then
        ADR_ACK_DELAY = ADR_ACK_DELAY * 2;
        ADR_ACK_LIMIT = ADR_ACK_LIMIT * 2;
    end
end
```

Table 3
Simulation parameters.

Parameters	Value
Simulation Time	8 days
Warm Up Time	1 day
Area Dimensions	40 Km × 40 Km
Gateways	5, 12
end-devices	20
Speed of End-Devices	0 mps–12 mps (step 3 mps)
Stop Duration	600 s–36000 s
Initial SF	SF9
Initial TP	8 dBm
Packet Size	20 bytes
Path Loss Model	LoRaLogNormalShadowing
Mobility Model	RandomWaypointMobility
n	2.08
σ	3.57
d_0	1000 m

5. System model

For the evaluation of our proposed approaches, we used the OMNET++ simulator and the INET framework. In addition, for the implementation of the LoRaWAN protocol, we used the FLoRa (Framework for LoRa) [15], that supports the LoRa physical layer, the LoRaWAN MAC protocol, and all necessary network elements such as gateways and networks servers. We have updated FLoRa to fulfil the requirements of LoRaWAN specifications of v1.1.

To evaluate our proposed algorithms, we have tested five different simulations setups with a different number of gateways and values of the speed of end-devices or the time interval between the packets transmitted by end-devices. We ran ten times each distinct simulation for specific parameters and took the average results. The simulation time for each one was set to 8 days, with 1 day as a warm-up period. During the simulations, each end-device moves in a random direction into the space of the simulation area. In the last of the simulations, each end-device stops for a random time after it reaches its destination before it continues its movement.

The simulation area is a rectangular area of dimensions of 40 km on both axes. We have randomly deployed the gateways and placed them in the middle 80% part of the simulation area. Moreover, we have prevented placing nearby two or more gateways. We are using 20 end-devices for all of our simulations, each one sending messages in a specific time interval and moving at a specific speed. Since we are not examining packet collisions, we think that this number is sufficient to evaluate the behaviour of our proposed mechanisms. Under these specifications, we have simulated five different scenarios described in detail in the following paragraphs.

The general parameters of the simulation setups are shown in Table 3.

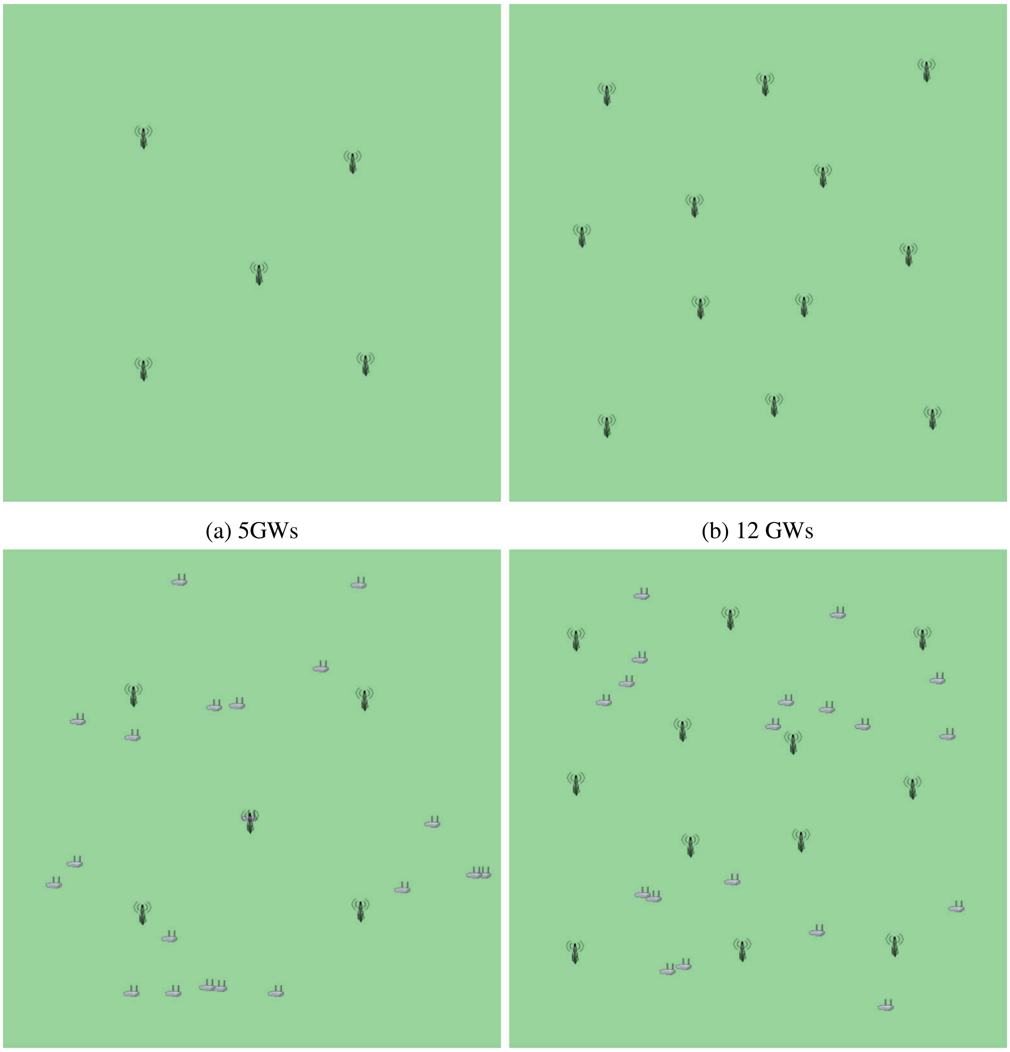
As a path loss model, we have used the LoRaLogNormalShadowing, which is implemented in FLoRa and based on LogNormal-Shadowing of INET. Eq. (10) defines the model.

$$PL_d = PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_\sigma \tag{10}$$

Where PL_d is the path loss in decibels (dB) at a distance d from the gateway. PL_0 is the path loss at a reference distance of d_0, while d_0 can take different values to make the model suitable for large or small areas. As described in [27], a value of 1 m to 10 m for d_0 is suitable for small areas, while a value of 1 km is suitable for large areas. Thus, for our implementation, we are using a value of 1 km for d_0. Finally, the parameter n is the path loss exponent, and the parameter X_σ represents the noise between the transmitter and the receiver that occurred by obstacles like mountains or buildings in the case of large areas. In this case, X_σ has a Gaussian distribution with a standard deviation σ in decibels (dB).

As a mobility model for the end-devices, we have used the RandomWaypointMobility model implemented in INET. Based on this model, each node moves in line segments, at a random destination and a random speed. Between each line segment, each node can wait for an arbitrary time. The minimum and maximum value of the speed and the minimum and maximum wait duration can be initialised based on our needs. We used a minimum speed of 0 mps (metres per second) for our implementation when we want stationary end-devices. In addition, in most scenarios, we used fixed velocity with values 3 mps, 6 mps, 9 mps and 12 mps. We did not use wait points between movements in line segments in these scenarios since we want to evaluate our implementation independently for different speeds. The last simulation tries to be closer to a real world scenario, where end-devices are moving with random speed, at a random wait point and make stops before continuing their movement. In this scenario, we used all the features that RandomWaypointMobility offers.

For the two of the simulations, we have deployed five gateways, while for the other three, we have deployed twelve gateways in the area in order to evaluate the response of our mechanisms in different levels of coverage of the field. An example of a deployment with five gateways is displayed in Fig. 4(a), where one gateway is randomly deployed close to the centre of the area while the rest of them are randomly deployed close to the centre of the four quadrants of the area. We selected this kind of deployment to avoid positioning two or more gateways close to each other. Similarly, Fig. 4(b) shows a setup with 12 gateways, which are deployed



(a) 5GWs (b) 12 GWs

(c) 5 GWs with 20 EDs (d) 12 GWs with 20 EDs

Fig. 4. Different simulation deployment examples.

as follows. Firstly, we logically divided the area into nine squares and then placed eight of the gateways close to the centre of the perimeter squares while placed the rest four gateways close to the corners of the central square. All of our simulations consider an area of 40 km in both axes, and a number of 20 end-devices at a random position was deployed. According to this, Fig. 4(c) presents the initial state of the setup with 5 gateways and random deployment of 20 EDs. In addition, Fig. 4(d) show the initial setup with 12 GWs and random deployment of 20 EDs.

6. Simulations results

This section presents and discusses the simulation results of our proposed LR-ADR and LR+ADR mechanisms. We are presenting and comparing them with four alternative ADR mechanisms, original ADR, ADR+, EMA-ADR, and G-ADR. The evaluation shows that both of our approaches are able to deliver a higher Packet Delivery Ratio (PDR) while keeping the Energy Consumption per Packet Delivered (ECPD) at low levels.

Moreover, we have used simulations with a difference in velocity of the end-devices while keeping the packet send time interval to a fixed value. In addition, we perform simulations with a specific speed while the time interval varies. Finally, we present a simulation where end-devices have a random velocity between specific boundaries, and they also make stops between their movements. In addition, the time interval between the packet transmissions of end-devices was kept fixed at this particular simulation.

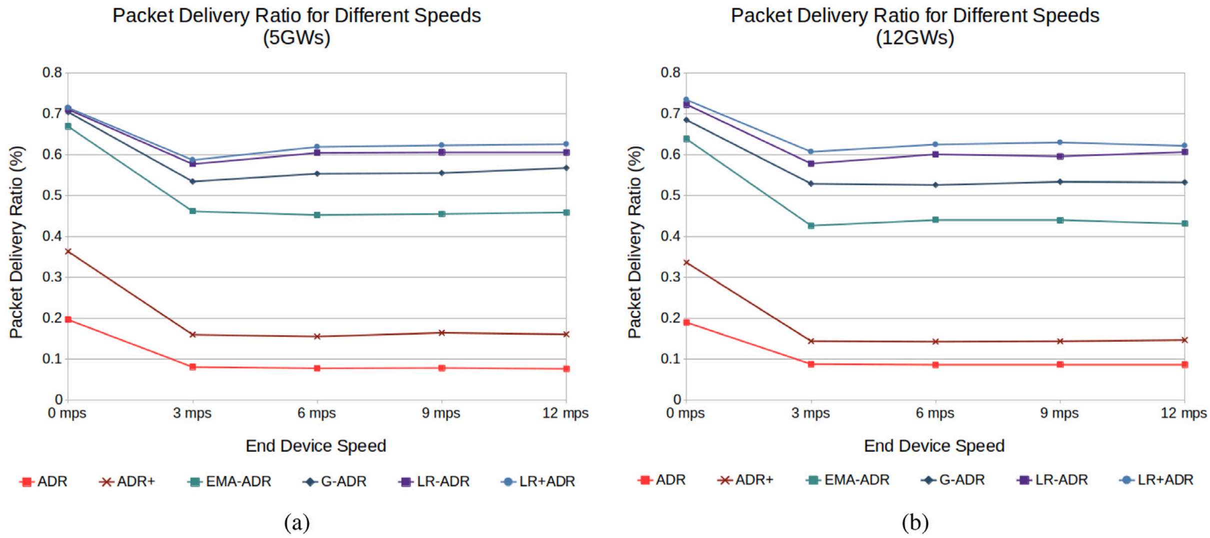


Fig. 5. Packet Delivery Ratio for Different Speeds of end-devices.

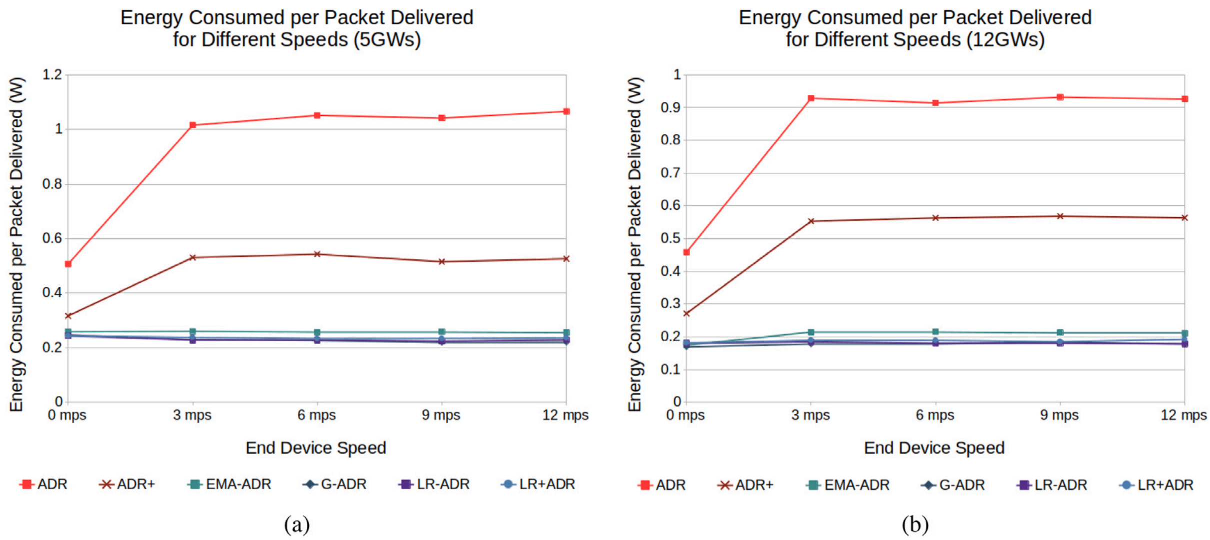


Fig. 6. Energy Consumed per Packet Delivered for Different Speeds of end-devices.

In our first evaluation, we deployed five gateways in the simulation area with 20 EDs. We have conducted five different simulations with different velocity for EDs. In the first simulation, all EDs were stationary, while in the next four, each EDs had a fixed velocity with 3 mps, 6 mps, 9 mps, and 12 mps, respectively. Each ED sent a packet periodically with a time interval of 240 s. Fig. 5(a) presents the comparison for the Packet Delivery Ratio (PDR) of all six ADR mechanisms. The results show that our LR-ADR approach is better than the competitors, while LR+ADR adds extra improvement to the PDR. In the same way, in Fig. 5(b) we are presenting the comparison in an area where we deployed 12 gateways in the field while the rest features were the same. In both setups, all mechanism have better performance when the end-devices were stationary. A close comparison between the two setups shows that the LR+ADR has a slight improvement when we have 12 GWs, while LR-ADR is at the same level. The other four mechanisms seem to perform lower when we deploy more gateways.

In Figs. 6(a) and 6(b) we are presenting the results for the Energy Consumption per Packet Delivered (ECPD). More specifically, we calculate and present the ratio of the total energy consumed from end-devices with the number of the total successfully delivered packets at the Network Server. This value shows the energy efficiency of each mechanism. The comparison shows that LR-ADR and

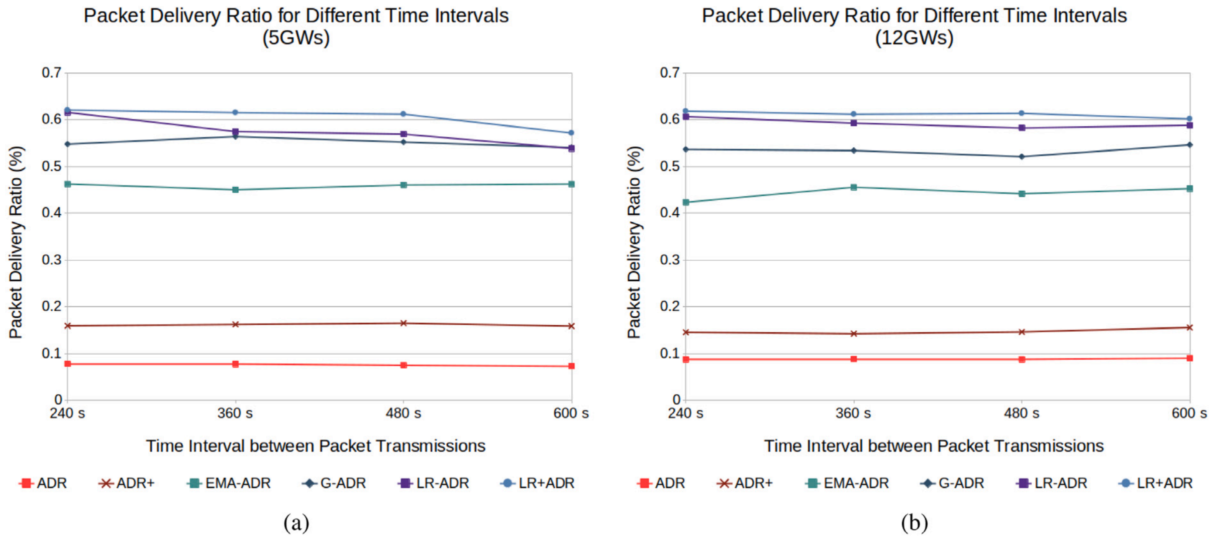


Fig. 7. Packet Delivery Ratio for different time interval between packets transmissions.

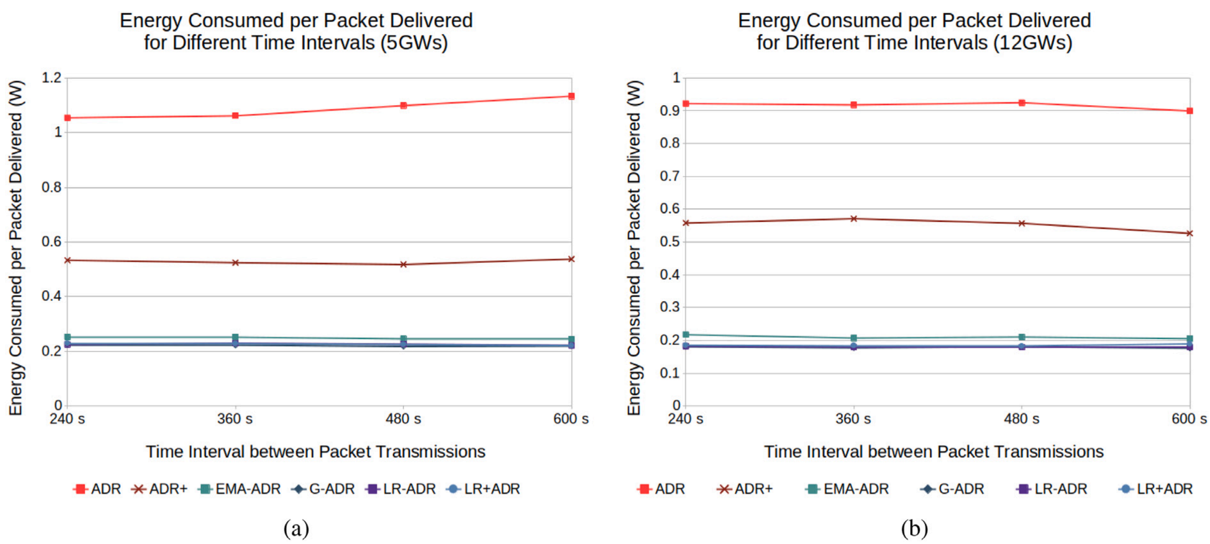


Fig. 8. Energy Consumed per Packet Delivered for different time interval between packets transmissions.

LR+ADR consume the same energy as the G-ADR while EMA-ADR consumes slightly more energy. In addition, ADR and ADR+ are far behind the competition.

The next setup of our simulations tries to evaluate how the time interval between packet transmission affects our proposed mechanisms. For this purpose, we keep the same velocity at 6 mps while we are changing the packet transmission time interval. More specifically, we have used the values 240 s, 360 s, 480 s, 600 s as the time interval between packet transmissions. Figs. 7(a) and 7(b) show the comparison between all competitors for deployments with 5 and 12 GWs respectively. The deployment with 5 GWs LR-ADR has slightly better performance than G-ADR, while LR+ADR gives a better improvement. In addition, in the 12 GWs deployment, they both seem to be more stable and give better performance than G-ADR, while EMA-ADR provides an even lower Packet Delivery Ratio.

Figs. 8(a) and 8(b) present a comparison of all mechanisms regarding the total energy consumption per successfully packets delivered. Both LR-ADR and LR+ADR are at the same level as G-ADR, while EMA-ADR is close but with slightly worst values. Finally, ADR and ADR+ consume more energy when comparing their performance with the total number of packets delivered.

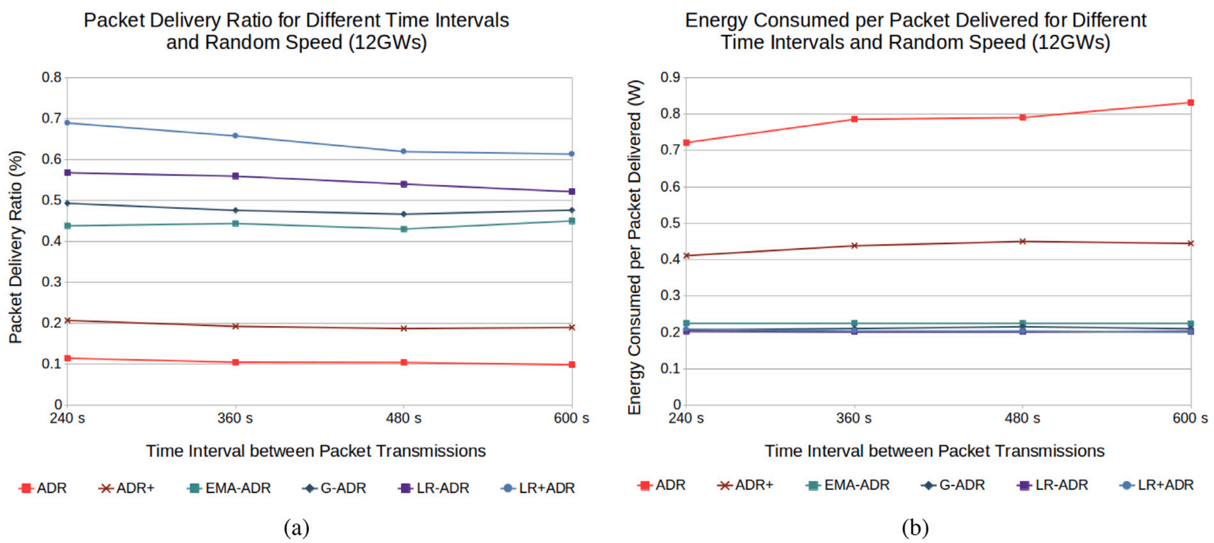


Fig. 9. Packet Delivery Ratio and Energy Consumed per Packet Delivered for different time interval between packets transmissions for end-devices moving with random speeds and stops.

Table 4
SF distribution for all mechanisms (PS = Packets Sent, PDR = Packet Delivery Ratio).

	SF7		SF8		SF9		SF10		SF11		SF12	
	PS	PDR	PS	PDR	PS	PDR	PS	PDR	PS	PDR	PS	PDR
ADR	47153	0.09	2686	0.38	540	0.59	26	0.58	0	0	0	0
ADR+	43410	0.17	5776	0.4	1139	0.58	112	0.7	55	0.18	9	0.09
EMA-ADR	27635	0.24	10343	0.53	7752	0.73	3425	0.88	987	0.95	321	0.78
G-ADR	23021	0.26	12271	0.54	9468	0.74	4225	0.9	1125	0.95	309	0.88
LR-ADR	19048	0.27	10688	0.57	10151	0.75	6186	0.91	2342	0.96	1943	0.99
LR+ADR	12831	0.37	8755	0.6	10363	0.75	8079	0.88	3963	0.93	6451	0.96

For the final evaluation, we used a deployment that tries to simulate a real world scenario. More specifically, we deployed 12 GWs in the area where all end-devices move with a random speed at a random wait point, waiting there before they continue their movement. Each stop takes at least 5 min or a maximum of 10 h. In Fig. 9(a), we are giving a comparison of the Packet Delivery Ratio between all competitors for different time intervals between packet transmissions. The results show that our approaches are above the competition providing better PDR. Finally, Fig. 9(b) presents the total energy consumption compared with the successfully received packets from the Network Server. Once again, LR-ADR and LR+ADR are at the same level as G-ADR, while EMA-ADR follows.

Table 4 shows the SF distribution for all mechanisms. The results correspond only to the simulation with time interval between packets transmission equal to 240 s and random speed and stops of the end-devices.

Two columns for each SF are displayed; the first one shows the number of packets sent (PS) for the corresponding SF, while the second one shows the Packet Delivery Ratio (PDR) for the same SF of each mechanism. The comparison shows that LR-ADR and LR+ADR use less packets with low SF values to achieve better performance. But even in low SF values, they reach better PDR than the competitors.

7. Conclusion

This paper proposed two solutions to enhance the original ADR mechanism and support LoRa mobile end-devices more efficiently. The aim was to improve the transmission reliability of the mobile end-devices with low moving speed and keep energy consumption as low as possible.

Our approach deals with the ADR mechanism and is twofold. First, we introduced the LR-ADR algorithm that enhances the mechanism from the Network Server side by presenting an improved algorithm considering the previous SNR values from the gateways and predicting the next expected SNR values for a specific end-device. For this purpose, we used simple linear regression for the last 10 SNR values for each particular gateway. Then, we predict the next expected SNR based on the linear regression function, and the average value for all gateways is added in a list for the next steps. When the end-device sends a message with the

ADRACKReq bit enabled, we calculate the average of the last 20 SNR values of the list and use it to select the optimal values for SF and TP in the same way that the ADR algorithm does.

Secondly, we proposed the LR+ADR algorithm, an adaptive improvement method in the end-device side, to quickly regain connectivity with the Network Server, if it continuously loses it. More specifically, we added the parameter ADR_ACK_RECEIVE_CNT to count the consecutively received ACK commands for the packets sent from an end-device. This parameter is used to detect if we have a reliable connection or not with the Network Server. Depending on this, we are decreasing the values of ADR_ACK_LIMIT, and ADR_ACK_DELAY in order to restore the connection faster if the end-device lost it. Instead, we are increasing them if consecutively acknowledgement commands return from the Network Server.

Our evaluation based on simulation modelling has proven that our approaches present better results than the existing proposed solutions, especially when the end-device enhancement is adopted. We have presented results for various moving speeds and different packet transmission intervals, comparing our proposals with four other solutions: the original ADR mechanism, ADR+, EMA-ADR and, G-ADR. The conducted evaluation focused on examining the Packet Delivery Ratio and total Energy Consumption per Packet Delivered on the Network Server. For mobile end-devices, the simulation results showed an improvement for Packet Delivery Ratio while keeping the Energy Consumption per Packet Delivered at low levels. In particular, in the scenario that mimics real world conditions, our schemes present an increase of up to 520% for PDR compared to the original ADR and an improvement of up to 38% compared to the best competitor (G-ADR). Moreover, they reduce ECPD up to 74% compared to the original ADR, while keeping it at the same level with the best competitor (G-ADR). In essence, our approaches are selecting more efficiently SF values to use in order to achieve better performance. Finally, the proposed schemes have backward compatibility with the original ADR mechanism, even if used only on the Network Server or the end-device side.

It becomes evident that the existing ADR schemes have the potential to efficiently support LoRa enabled mobile end-devices and more research efforts are required to this direction. Altering the mechanism from the NS side by trying to predict the next position of the end-device may also help. In addition, enhancing the mechanism from the end-devices side allows them to decide on their own strategy and further improve reliability. However, this should be done, considering all the restrictions that end-devices have. In this vein, the aforementioned approaches constitute future directions of this work, along with further refining the prediction for the next transmission state of the mobile devices.

References

[1] L. Sciullo, F. Fossemo, A. Trotta, M. Di Felice, Locate: A lora-based mobile emergency management system, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 1–7.

[2] M.G. Ikhsan, M.Y.A. Saputro, D.A. Arji, R. Harwahyu, R.F. Sari, Mobile LoRa gateway for smart livestock monitoring system, in: 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS), IEEE, 2018, pp. 46–51.

[3] P. Mayer, M. Magno, A. Berger, L. Benini, RTK-LoRa: High-precision, long-range, and energy-efficient localization for mobile IoT devices, IEEE Trans. Instrum. Meas. 70 (2020) 1–11.

[4] N. Zinas, S. Kontogiannis, G. Kokkonis, S. Valsamidis, I. Kazanidis, Proposed open source architecture for long range monitoring. The case study of cattle tracking at Pogoniani, in: Proceedings of the 21st Pan-Hellenic Conference on Informatics, 2017, pp. 1–6.

[5] L.-Y. Chen, H.-S. Huang, C.-J. Wu, Y.-T. Tsai, Y.-S. Chang, A lora-based air quality monitor on unmanned aerial vehicle for smart city, in: 2018 International Conference on System Science and Engineering (ICSSE), IEEE, 2018, pp. 1–5.

[6] Y. Cheung, M. Qiu, M. Liu, Autonomous vehicle communication in v2x network with lora protocol, in: International Conference on Smart Computing and Communication, Springer, 2019, pp. 398–410.

[7] N. Benkahla, H. Tounsi, S. Ye-Qiong, M. Frikha, Enhanced ADR for LoRaWAN networks with mobility, in: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2019, pp. 1–6.

[8] A. Farhad, D.-H. Kim, S. Subedi, J.-Y. Pyun, Enhanced LoRaWAN adaptive data rate for mobile internet of things devices, Sensors 20 (22) (2020) 6466.

[9] C. Bouras, A. Gkamas, S.A.K. Salgado, Energy efficient mechanism for LoRa networks, Internet of Things 13 (2021) 100360.

[10] M. Stellin, S. Sabino, A. Grilo, LoRaWAN networking in mobile scenarios using a WiFi mesh of UAV gateways, Electronics 9 (4) (2020) 630.

[11] P.-Y. Chen, L. Bhatia, R. Kolcun, D. Boyle, J.A. McCann, Contact-aware opportunistic data forwarding in disconnected lorawan mobile networks, 2020, arXiv preprint arXiv:2004.06614.

[12] J. Park, K. Park, H. Bae, C.-K. Kim, EARN: Enhanced ADR with coding rate adaptation in LoRaWAN, IEEE Internet Things J. 7 (12) (2020) 11873–11883.

[13] D.-Y. Kim, S. Kim, H. Hassan, J.H. Park, Adaptive data rate control in low power wide area networks for long range IoT services, J. Comput. Sci. 22 (2017) 171–178.

[14] J. Finnegan, R. Farrell, S. Brown, Analysis and enhancement of the LoRaWAN adaptive data rate scheme, IEEE Internet Things J. 7 (8) (2020) 7171–7180.

[15] M. Slabicki, G. Premsankar, M. Di Francesco, Adaptive configuration of LoRa networks for dense IoT deployments, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–9.

[16] A. Raychowdhury, A. Pramanik, Survey on LoRa technology: Solution for Internet of Things, Intell. Syst. Technol. Appl. (2020) 259–271.

[17] S.S. Reka, B.K. Chezian, S.S. Chandra, A novel approach of iot-based smart greenhouse farming system, in: Green Buildings and Sustainable Engineering, Springer, 2019, pp. 227–235.

[18] W. Hassan, M. Føre, J.B. Ulvund, J.A. Alfredsen, Internet of fish: Integration of acoustic telemetry with LPWAN for efficient real-time monitoring of fish in marine farms, Comput. Electron. Agric. 163 (2019) 104850.

[19] F.V.M. de Souza, R.D.S. Rabello, LoRaWan: Low cost solution for smart cities, in: Smart and Digital Cities, Springer, 2019, pp. 181–193.

[20] D. Bruneo, S. Distefano, M. Giacobbe, A.L. Minnolo, F. Longo, G. Merlino, D. Mulfari, A. Panarello, G. Patanè, A. Puliafito, et al., An iot service ecosystem for smart cities: The# smartme project, Internet of Things 5 (2019) 12–33.

[21] J.M. Paredes-Parra, A.J. García-Sánchez, A. Mateo-Aroca, A. Molina-García, An alternative Internet-of-Things solution based on LoRa for PV power plants: data monitoring and management, Energies 12 (5) (2019) 881.

[22] L. Oliveira, J.J. Rodrigues, S.A. Kozlov, R.A. Rabêlo, V.H.C.d. Albuquerque, MAC layer protocols for Internet of Things: A survey, Future Internet 11 (1) (2019) 16.

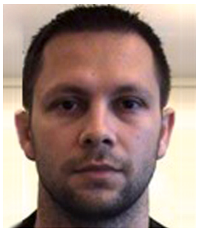
[23] M.C. Bor, U. Roedig, T. Voigt, J.M. Alonso, Do LoRa low-power wide-area networks scale?, in: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2016, pp. 59–67.

[24] J.C. Liando, A. Gamage, A.W. Tengourtius, M. Li, Known and unknown facts of LoRa: Experiences from a large-scale measurement study, ACM Trans. Sensor Netw. 15 (2) (2019) 1–35.

[25] S. Li, U. Raza, A. Khan, How agile is the adaptive data rate mechanism of LoRaWAN? in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 206–212.
[26] Y. Zhang, L. Ye, Indoor localization method based on AP and local linear regression algorithm, in: 2017 IEEE 17th International Conference on Communication Technology (ICCT), IEEE, 2017, pp. 1122–1126.
[27] O. Dieng, C. Pham, O. Thiare, Comparing and adapting propagation models for LoRa networks, in: 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)(50308), IEEE, 2020, pp. 1–7.

Vasileios Moysiadis received the Diploma degree in Computer Science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1998, and the M.Sc. degree in Applied Informatics in Computer Systems from the University of Macedonia, Thessaloniki, Greece, in 2008. Currently, he is a Ph.D. student in the Department of Electrical and Computer Engineering at the University of Western Macedonia, Kozani, Greece. He is currently involved as a Research Associate in national and European funded research projects. His research interests are in the area of Low Power Wireless Area Networks, Internet of Things, Smart Farming, and Fog Computing.

Dr. Thomas Lagkas is Assistant Professor of the Department of Computer Science of the International Hellenic University. He received his Ph.D. in computer science from the Aristotle University of Thessaloniki, Greece, in 2006. He has been Lecturer and then Senior Lecturer of The University of Sheffield International Faculty — CITY College, from 2012 to 2019. He also served as Research Director of the Computer Science Department of CITY College and Leader of the ICT Track of the South-East European Research Centre. His research interests are in the areas of IoT communications and distributed architectures, communication security, wireless communication networks, QoS in medium access control, mobile multimedia communications, power saving/fairness ensure for resource allocation in wireless sensor-cooperative broadband networks as well as in hybrid Fibre-Wireless networks, e-health data monitoring, 5G and beyond systems, flying ad hoc networks, and computer-based educational technologies with more than 90 publications at a number of widely recognised international scientific journals and conferences. He also participates in the Editorial Boards of the IEEE IoT Journal, the Elsevier Computer Networks Journal, the Springer Telecommunication Systems Journal, and the EURASIP Journal on Wireless Communications and Networking (published by Springer). Moreover, he actively participates in drafting research funding proposals, as well as in the implementation of the corresponding EU projects (namely, ELECTRON, TERMINET, SDN-microSENSE, SmartROOT, JAUNTY, FuseGI, ARRANGE-ICT, et al.).



Prof. Vasilis Argyriou received a B.Sc. degree in computer science from Aristotle University of Thessaloniki, Greece, in 2001 and the MSc and Ph.D. degrees from the University of Surrey, in 2003 and 2006, respectively, both in electrical engineering working on registration. Vasilis joined the Communications and Signal Processing (CSP) Department, Imperial College, London, in 2007 where he was a Research Fellow working on 3-D object reconstruction. Now, he is a Professor at Kingston University working on computer vision and AI for crowd and human behaviour analysis, computer games, entertainment and medical applications. Also research is conducted on educational games and on HCI for augmented and virtual reality (AR/VR) systems. Over the last 5 years he participated in many EU H2020 and UK projects including direct collaborations with industrial partners. The projects are mainly focused on scene analysis, pedestrian detection, drones, AR/VR for training and data analytics using advanced deep learning architectures and computer vision solutions. Currently he is part of the RoViT Group at Kingston University and is collaborating with other institutions in Greece, Cyprus and Spain.

Dr. Antonios Sarigiannidis received the B.S. and M.S. degrees in computer science from the Department of Informatics, Aristotle University, Thessaloniki, Greece, in 2007 and 2009, respectively. Also, he received the Ph.D. degree from the same department in 2016. His research interests include telecommunication networks, internet of things, cybersecurity, and smart grid. He is author or coauthor of more than 20 journal paper, conference papers, and chapter books. He has been involved in many national and European RD projects. He received the CCNA Routing and Switching certification from Cisco in 2018. He is the co-founder of Sidroco Holdings Ltd, Nicosia, Cyprus.

Prof. Ioannis D. Moscholios was born in Athens, Greece, in 1976. He received the Dipl. Eng. degree in Electrical Computer Engineering from the University of Patras, Patras, Greece, in 1999, the M.Sc. degree in Spacecraft Technology Satellite Communications from the University College London, UK, in 2000 and the Ph.D. degree in Electrical Computer Engineering from the University of Patras, in 2005. From 2005 to 2009 he was a Research Associate at the Wire Communications Laboratory, Dept. of Electrical Computer Engineering, University of Patras. From 2009 to 2013 he was a Lecturer in the Dept. of Telecommunications Science and Technology, University of Peloponnese, Tripolis, Greece. From 2013 to 2018 he was an Assistant Professor in the Dept. of Informatics and Telecommunications, University of Peloponnese, Tripolis, Greece. Currently, he is an Associate Professor in the Dept. of Informatics Telecommunications, University of Peloponnese, Tripolis, Greece. His research interests include teletraffic engineering, simulation and performance analysis of communication networks. He has published over 175 papers in international journals/conferences and is a co-author of the book: Efficient Multirate Teletraffic Loss Models Beyond Erlang (IEEE Press, Wiley, April 2019). He has served as a Guest Editor in: (a) IET Communications, (b) IET Networks, (c) Applied Sciences and d) Mobile Information Systems. He has also served as an Associate Editor in IEICE Transactions on Communications. He is an IARIA Fellow and a member of the Technical Chamber of Greece (TEE).

Prof. Panagiotis Sarigiannidis is an Associate Professor in the Department of Electrical and Computer Engineering in the University of Western Macedonia, Kozani, Greece since 2016. He received the B.Sc. and Ph.D. degrees in computer science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2007, respectively. He has published over 180 papers in international journals, conferences and book chapters, including IEEE Communications Surveys and Tutorials, IEEE Transactions on Communications, IEEE Internet of Things, IEEE Transactions on Broadcasting, IEEE Systems Journal, IEEE Wireless Communications Magazine, IEEE/OSA Journal of Lightwave Technology, IEEE Access, and Computer Networks. He has been involved in several national, European and international projects. He is currently the project coordinator of three H2020 projects (SPEAR, EVIDENT, TERMINET) while he coordinates two operational project (MARS, SmartROOT). He also serves as a principal investigator in one H2020 project (SDN-microSENSE) and three more Erasmus+ KA2 projects (Arrange-ICT, JAUNTY, STRONG). His research interests include telecommunication networks, internet of things and network security. He is an IEEE member and participates in the Editorial Boards of various journals, including International Journal of Communication Systems and EURASIP Journal on Wireless Communications and Networking.