Emergency Communication in IoT Scenarios by Means of a Transparent LoRaWAN Enhancement

Emiliano Sisinni, *Member, IEEE*, Dhiego Fernandes Carvalho, *Graduate Student Member, IEEE*, and Paolo Ferrari, *Member, IEEE*

***Abstract*—This work deals with the management of sporadic and rare events linked with emergency situations in wireless Internet-of-Things (IoT) scenarios. The goal is to increase the performance of the emergency communication, when low-power wide area networks (LPWANs) are used as IoT backbone. In the proposed approach, a device usually operates as a normal node but, in case of emergency, can use the novel LoRa-REP access method. In this work, the LoRa-REP, based on message replication, is discussed focusing on its capability of reducing average transaction time and increasing success probability. Two operational paradigms have been considered and tested: public LoRaWAN infrastructure with cloud-based backend, and private LoRaWAN networks with local backend (edge/fog computing). Typical examples of public networks are smart cities, whereas local networks are often used in industry or building automation. Additionally, two real use cases (for public and local scenarios) are provided to show the effectiveness of the proposed approach. The experimental results (with a prototype device implementing LoRa-REP and named eNode) show that the success probability of the emergency communication can be increased up to 99.5%, and the average transaction time can be reduced up to 15% with respect to LoRaWAN without retries or up to 50% with respect to LoRaWAN with retries.**

***Index Terms*—Low-power wide area network (LPWAN), mixed criticality, priority communication, wireless sensor network.**

I. Introduction

The Internet of Things (IoT) is an umbrella term describing large interconnected systems, including sensor (and actuator) networks, typically exploiting wireless links and supporting fixed and mobile communications. Due to the many advantages such an infrastructure can offer, it has become a hot research topic in the recent past. Very diverse application fields have been suggested. For instance, one of the most important IoT successes, Industry 4.0, shows large improvements in terms of manufacturing efficiency and waste reduction obtained by means of continuous monitoring/control of processes and products [1]. Connection with the Internet in an IoT scenario can be obtained with very assorted networks, but the majority of the end nodes use wireless communications [2].

Manuscript received April 8, 2020; revised June 25, 2020; accepted July 17, 2020. Date of publication July 22, 2020; date of current version October 9, 2020. *(Corresponding author: Emiliano Sisinni.)*

The authors are with the Department of Information Engineering, University of Brescia, 25123 Brescia, Italy (e-mail: emiliano.sisinni@unibs.it).

Digital Object Identifier 10.1109/JIOT.2020.3011262

As a matter of fact, most of the available literature is mainly focused on effective wireless infrastructure implementation, able to support low-cost, low-power consumption nodes, in order to ensure long lifetime for tiny, battery-powered devices. On the other hand, less attention has been paid to the timeliness of data exchanges and communication reliability [3].

However, with the expansion and diversification of the application scenarios, a considerable amount of data packets need to be sent to the final destination as soon as possible, often requiring a confirmation of the correct reception. This requirement is common to packets carrying emergency data, e.g., the alarm information in the medical rescue service, the fire information in forest fire monitoring service, etc. As a consequence, proper management of emergency data packets, and their acknowledgment, has become a serious challenge.

In this work, the authors consider the largely diffused low-power wide area network (LPWAN) solutions [4], originally designed for smart cities applications, where data from smart objects (e.g., smart meters) have to be sporadically sent to a data sink in the cloud. In particular, the focus is about the LoRaWAN technology, which has been proved to be effective and can boast a very large number of deployments. However, as aforementioned, timely message delivery and resilience are somehow overshadowed by concerns about power consumption and low cost, limiting the use of LoRaWAN in some application fields [5]. To overcome these limits, authors develop an innovative message replication mechanism, called LoRa-REP, fully compliant with the LoRaWAN specifications and ensuring investment protection. In particular, in this work, authors:

1) identify and implement two different real-world scenarios for experimental evaluation: the first one is inspired by smart city applications and it has a public and cloud-based backend; the second one is inspired by industrial situations and has local and private backend;
2) report results of a vast measurement campaign, with fixed and mobile devices based on purposely designed eNodes, aiming at evaluating obtainable performance in terms of communication resilience and timing performance.

This article is structured as in the following. Section II briefly summarizes LoRaWAN concepts. Section III introduces the LoRa-REP protocol and Section IV describes the eNode. Section V introduces the experimental model, Sections VI and VII present the experimental results.

II. LPWAN AND LORAWAN

The acronym LPWAN is generally used for addressing all the low bandwidth, low-data rate, and low-cost wireless technologies developed for IoT-like applications. However, the different technologies under this heading can be quite dissimilar, falling into two groups. The first one includes standard cellular connectivity operating in licensed bands; most common are narrow band-Internet of Things NB-IoT, and LTE Cat M [6]. The second one contains technologies that mimic mobile connectivity but operate in unlicensed frequencies. Currently, the most famous is LoRaWAN, based on the proprietary LoRa radio developed by Semtech. In this work, we specifically address LoRaWAN, due to the many networks already deployed, especially in the smart city context.

A. LoRaWAN: PHY and MAC

The proprietary LoRa physical layer (PHY) [7] exploits the chirp spread spectrum modulation, enabling the coding of SF (the Spreading Factor) bits per chirp. Indeed, the chirp duration T_C is divided into 2^{SF} discrete-time positions, each one hosting a possible "jump" in the chirp frequency trajectory. On the contrary, the chirp bandwidth $BW = \mu T_C$ is fixed and the chirp rate is $\mu = (2^{SF}/T_C{}^2)(Hz/s)$. In light of this, the raw bit rate is $Rb = SF(BW/2^{SF}) \cdot CR$ (bps), considering that forward error correction (FEC) with coding rate CR is further applied. As a consequence, using higher SF values permits a tradeoff between the noise immunity (increased thanks to the higher processing gain of longer chirps) and the data rate (decreased for longer chirps). Moreover, due to the quasiorthogonal behavior of chirps having different durations, each SF defines a sort of "virtual" channel, compensating for limited bandwidth in sub-GHz unlicensed spectrum regions and permitting for some forms of adaptive data rate (ADR) strategies. Finally, since the higher processing gain of larger SFs results in better sensitivity at the receiver side, a tradeoff between the data rate and the coverage exists as well.

Regarding the medium access control (MAC) strategy, pure ALOHA is considered in LoRaWAN to reduce, as much as possible, the implementation complexity. An additional advantage is the reduction of MAC control messages, which in turn reduces delay and energy consumption. However, so-called channel activity detection (CAD) can be used to look for LoRa preambles and to exploit some avoidance techniques. Data exchange is bidirectional, arranged into uplink and downlink (i.e., from the field and toward the field, respectively). Nevertheless, since the main target applications are for "monitoring," uplink is of most importance and downlinks are usually minimized (in order to reduce the overall traffic and to follow duty-cycle constraints). Accordingly, in most cases, the communication is event based: the device sends an uplink and, subsequently, two mutually exclusive downlink windows (RX1 and RX2) are available. Specifications require that communication parameters in RX1 are derived from the corresponding uplink (same channel, while the SF may have an offset). RX2 leverages on the fixed channel and SF values (e.g., default parameters in EU are 869.525 MHz/DR0-SF12 @ BW = 125 kHz), so that it could be set to be always orthogonal to RX1.

B. LoRaWAN: Architecture

LoRaWAN exploits a sort of two-tiered architecture, as shown in Fig. 5, in which the experimental testbed is described. The first one consists of single-hop (i.e., star topology) wireless links between field devices and the surrounding gateways (GWs). Each GW behaves as a simplified base station in mobile communications, forwarding messages to/from the wired backend. In order to keep the infrastructure implementation as cheap as possible, messages are opaque to the GWs, which only tunnel them into the backhaul. The backend is the second tier and consists of several (logical) servers typically interconnected by means of an IP-based network. In other words, backend can be both local (i.e., based on an Intranet) or located in the cloud (i.e., based on Internet). In particular, the GWs are connected to so-called network server (NS, managing the overall network behavior); the NS is connected to at least one application server (AS, which permits the integration with the end-user applications, typically via a message-oriented middleware as an MQTT Broker). The latest specifications introduce the join server (JS) as well, in order to decouple the affiliation from the network management, as happens in previous releases. More details about activation procedures and security aspects, which do not affect the proposed strategy, can be found in [8].

C. Related Works

The very broad adoption of LoRaWAN is testified by the interest of telecommunications providers (e.g., the Dutch telecommunications operator KPN, and The Things Network TTN), multiutility companies, and research institutes. In turn, this situation motivates the large number of works that can be found in the literature. In particular, obtainable performance and limitations of the standard have been deeply analyzed. Particularly interesting, in light of the strategy adopted in the proposed approach, is the idea of improving reliability by means of replications, e.g., as suggested in [9] and [10], where evaluation is limited to simulations. Another important aspect is the impact that downlink traffic (including acknowledges of uplink messages) can have on the overall network throughput [11]–[13]. For this reason, downlink usage should be limited only if really required by the application. As a matter of fact, the intended IoT scenario privileges uplink, but may suffer from simultaneous activation of several nodes "sensible" to the same event, e.g., as pointed out in [14]; in such a case, transmission should be properly randomized, e.g., retries should happen after a certain backoff delay.

Anyway, in the survey [12] on confirmed traffic, a more deep discussion about the retry mechanism of LoRaWAN is reported. It highlights a lack in the specifications about retries, resulting in not unique interpretation so that "much of this [retry] procedure is left to the developer of the end device and NS."

III. Proposed Extension of LoRaWAN for Alarm and Emergency Situations

In this section, a brief overview of alarm management is provided in order to highlight the most important requirements in terms of communication reliability and real-time behavior during emergency situations. Subsequently, the proposed LoRa-REP strategy is introduced, which permits to increase both resilience and timeliness of LoRaWAN communications. Notably, with the proposed approach, the legacy investments are preserved thanks to full compatibility with the original standard specifications.

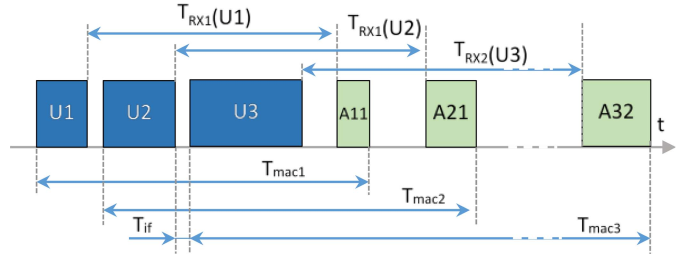

Fig. 1. LoRa-REP replication schema (example with three replicas).

A. Management of Alarms and Emergency

Alarm management systems are well known in the process control industry, where they are strictly coded by international norms. The most important characteristics of alarms are: 1) prioritization, i.e., alarms should be addressed according to their criticality level; 2) timeliness, i.e., alarms must be signaled on time; and 3) requiring response, i.e., alarms should be acknowledged in order to specify that a proper response is started.

Nowadays, alarms and acknowledgments in a plant are transferred via digital communication systems (such as fieldbuses), thus the aforementioned characteristics deeply affect requirements in terms of data exchange robustness and mixed-criticality capability, since the same medium is used for transferring information with different "importance" [15], [16]. Wireless solutions have been demonstrated to be effective thanks to their flexibility [17]. Generally speaking, resilience is guaranteed by employing diversity and redundancy, whereas criticality is handled by means of different resource scheduling [18], [19]. As a matter of fact, both needs require purposely designed MAC protocols [20], [21]. Recently, alarm management in emergency conditions in complex systems (such as a whole smart city) emerged as a new research field, as demonstrated by [22] and [23].

Even if redundancy has been already used in some LoRaWAN applications, the proposed LoRa-REP replication scheme satisfies the requirements of the previous examples managing both uplink and downlink, differently from [24], and in a simpler way than [25]. Its advantages in terms of reduced delays and better success probability are deeply discussed in the following section.

B. LoRa-REP Protocol

The LoRa-REP protocol coexists with LoRaWAN and it exploits diversity and redundancy for improving the reliability of emergency communication. It must be remembered that LoRaWAN specifications define local norms about maximum transmitting power and duty-cycle. If operated in Europe, the unlicensed 868-MHz region is organized in sub-bands by the recommendation 70-03, which is normed by the ETSI standard EN300220 (requiring the duty-cycle evaluation per sub-band on 1-h long interval). In particular, the observation interval shall be representative of the typical usage of the device, including 99% of transmissions generated during its operational lifetime. However, LoRa-REP transmissions do not contribute to the duty-cycle evaluation, since LoRa-REP-enabled nodes normally transmit following LoRaWAN rules, while a LoRa-REP transaction is generated only after a sporadic emergency event.

LoRa-REP relies on the use of different virtual channels (i.e., use of different SF values), which standard LoRaWAN has, to provide multiple redundant communication possibilities for nodes with emergency traffic. As stated in Section II-C, the LoRaWAN standard way of dealing with the repetition of messages is not completely clear, and, as a consequence, it is rarely used. LoRa-REP has been conceived for providing a fully compliant alternative to LoRaWAN with retries. In particular, LoRa-REP considers confirmed transactions (i.e., the application requires the uplink message to be followed by a downlink confirmation) because of the emergency nature of its payload. It has to be highlighted that increasing both communication resilience and the timeliness of confirmed transactions is of main concern. For this reason, LoRa-REP consider neither backoff delays between replicas nor any retransmission mechanism if all the programmed attempts fail. In case it is needed, retransmission strategy must be implemented at application level. This is a reasonable approach, since the "application" in the considered scenarios is an "emergency management" application and it may have also several other priorities to take into account.

The core idea of LoRa-REP is to replicate n times the emergency uplink frame – $Uj, j = 1 \cdots n$—using different SFs for each replica, as displayed in Fig. 1. The message copies are transmitted by "virtual nodes," which are configured as different devices at the data link level but share the same application level. Given n confirmed uplink, also the associated n downlink confirmations – $Ajk, j = 1 \cdots n$ and $k = 1, 2$—are expected as well (where k represents the RXk window, and it is decided by the NS). The uplink frames are transmitted from lower to higher SF values, i.e., from shorter to longer over-the-air durations. It should be noted that LoRa-REP does not take into account the signal strength of incoming messages for optimization, i.e., the scheduling is static.

LoRa-REP is oriented to be fast and simple, hence all the $Uj, j = 1 \cdots n$ must be sent before the RX1 window of U1. Therefore, there are constraints on both the number of permitted replicas and their durations. In particular, the following equation must hold, where T_{RX1} is the (fixed, but configurable) time distance between the end of an uplink message and the start of the corresponding RX1 window; T_{Uj} is the duration of the jth uplink frame (which depends on its SF), and T_{if} is

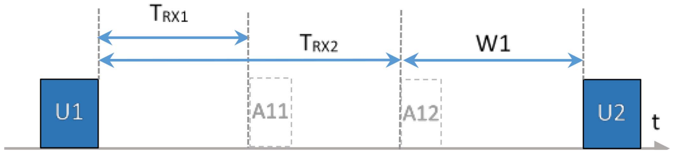

Fig. 2. Standard LoRaWAN replication schema.

the interframe space between successive frames:

$$(n-1)T_{if} + \sum_{j=2}^{n} T_{Uj} < T_{\mathrm{RX1}} \Rightarrow n < \frac{T_{\mathrm{RX1}} + T_{if} - \sum_{j=2}^{n} T_{Uj}}{T_{if}}. \quad (1)$$

It should be remarked that some restrictions may apply to the general case of (1) when constrained resource implementation of the LoRa-REP node are considered (see Section IV-A for details).

As previously stated, LoRa-REP is also aimed to complete an emergency transaction (i.e., to transmit high priority message followed by a confirmation) in the least amount of time. Consequently, a significant metric is the transaction delay D_T. The minimum value $D_{T,\min}$ is possible when A11 (i.e., the first RX1 downlink) is correctly received, as in (2), where T_{A11} is the duration of the downlink frame

$$D_{T,\min} = T_{U1} + T_{\mathrm{RX1}} + T_{A11}. \quad (2)$$

However, LoRa-REP takes advantage from higher noise immunity of higher SFs, and the transaction could be completed even if some Uj and Ajk are lost. The maximum transaction delay $D_{T,\max}$ corresponds to the worst case scenario: the transaction is completed when An2 is received (i.e., the last downlink An in RX2). If T_{Ajk} is the duration of the downlink frame corresponding to the jth uplink in the receiving window k, the expression of $D_{T,\max}$ is

$$D_{T,\max} = T_{An2} + T_{\mathrm{RX2}} + (n-1)T_{if} + \sum_{j=1}^{n} T_{Uj}. \quad (3)$$

If the NS is configured to use only the RX1, the $D_{T,\max}$ is shorter, being

$$D_{T,\max} = T_{An1} + T_{\mathrm{RX1}} + (n-1)T_{if} + \sum_{j=1}^{n} T_{Uj}. \quad (4)$$

There is a direct advantage over LoRaWAN standard way of dealing with retries. The time diagram of standard LoRaWAN confirmed transaction is shown in Fig. 2.

In LoRaWAN, the reiteration of the message transmission (on different SFs) in case of unsuccessful transaction is possible. However, the LoRaWAN specifications require that if the downlink is not received neither in RX1 nor in RX2 windows following the Uj uplink, a new uplink transmission $Uj+1$ can occur only after the time interval Wj. In particular, Wj must have a random contribution, $Wj_{rnd} = \mathcal{U}[-1s, +1s]$, and must be long enough to satisfy duty-cycle constraint, so that

$$Wj = \begin{cases} 2s + Wj_{rnd}, & \text{if } T_{Uj}\left(\frac{1-DC}{DC}\right) < 2s + Wj_{rnd} \\ T_{Uj}\left(\frac{1-DC}{DC}\right) + Wj_{rnd}, & \text{otherwise.} \end{cases} \quad (5)$$

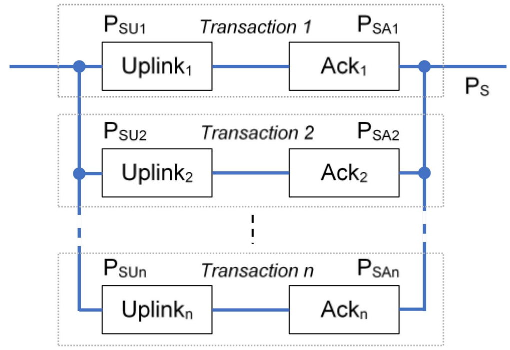

Fig. 3. LoRa-REP success probability model.

As a consequence, the maximum transaction delay for confirmed message in LoRaWAN, $D^L_{T,\max}$ is

$$D^L_{T,\max} = T_{An2} + T_{\mathrm{RX2}} + \sum_{j=1}^{n-1} (Wj + T_{\mathrm{RX2}}) + \sum_{j=1}^{n} T_{Uj}. \quad (6)$$

In detail, comparing (3) and (4) and (6), it can be observed that $\sum_{j=1}^{n-1}(Wj + T_{\mathrm{RX2}}) \gg (n-1)T_{if}$; thus, LoRa-REP has a lower maximum transaction delay. Additionally, the minimum transaction delay is fixed in LoRa-REP and it only depends on the minimum SF chosen for the message replicas, while in LoRaWAN the minimum delay increases as the SF increases.

C. Analysis of Success Probability

Another important metric for LoRa-REP is the transaction success probability. Without losing generality, the considered scenario includes nodes that usually transmit unconfirmed messages (thus not generating downlink messages), while they use confirmed message (i.e., with an acknowledgment) only in case of emergency (e.g., as supposed in [26]). Hence, the success probability of the proposed approach can be seen as the result of the model shown in Fig. 3. The success probability P_{Sj} of each parallel transaction is described by (7), where P_{SUj} is the probability for the GW to correctly receive the uplink Uj, and P_{SAj} is the probability for the node to correctly receive the downlink Aj

$$P_{Sj} = P_{SUj} \cdot P_{SAj}. \quad (7)$$

Note that the calculation of P_{SAj} depends on the NS strategy for transmitting the acknowledge. NS uses just one of the two downlink windows (mutually exclusive choice between RX1 or RX2). If the choice is statically configured in the NS, the probability of success is equal to the success probability of that windows: $P_{SAj} = P_{SAj1}$ or $P_{SAj} = P_{SAj2}$. Otherwise, if the choice is random $P_{SAj} = P_{\mathrm{RX1}} \cdot P_{SAj1} + (1 - P_{\mathrm{RX1}}) \cdot P_{SAj2}$, where the P_{RX1} is the probability of choosing RX1.

Considering that the messages transmitted using different SF values are orthogonal, and that the overall transaction is successful if at least one transaction is successful, the transaction success probability of LoRa-REP is given by

$$P_S = 1 - \prod_j \left(1 - P_{SUj} \cdot P_{SAj}\right). \quad (8)$$

It is clear that, in LoRa-REP, adding single transactions to the overall transaction results in an increase of the overall success probability.

It is important to underline that the success probability of LoRaWAN with retries is the same of LoRa-REP: also the LoRaWAN retry strategy is successful if at least one transaction is successful.

IV. Implementation of LoRa-REP Device

In this section, some considerations about the implementation of an end node with the LoRa-REP strategy are discussed. In particular, after a general description of the requirements for an engineered implementation, the prototyping platform developed by authors for speeding up on the field evaluation is introduced.

A. Mapping LoRa-REP on Real Devices

The LoRa-REP has been designed in order to avoid any violations of LoRaWAN, so that it can be implemented using any compliant radio. The only critical constrain is the need for $Aj1$ and $Aj2$ ($j = 1 \cdots n$) orthogonality to ensure correct reception. Radio devices embedded into GWs permit the simultaneous reception of frames transmitted using different channels and/or SFs. Therefore, it is enough to restrict the possible uplink configuration, reducing the overall communication opportunities offered by the specifications and norms, reserving some resources for RX2 transmissions only. On the contrary, when regular end device transceivers are considered (e.g., the single-chip solution SX1276 provided by Semtech), time overlapping of frames must be avoided. The easiest way to satisfy such a condition is to confine all the retries and the Aj1 messages into the $T_{\mathrm{RX2}} - T_{\mathrm{RX1}} = 1$ s interval, as in the following equation:

$$(N-1)T_{if} + \sum_{j=2}^{n} T_{Uj} + T_{An1} < T_{\mathrm{RX2}} - T_{\mathrm{RX1}}. \tag{9}$$

As a consequence, the number of possible replicas becomes as in the following equation:

$$n < \frac{1s + T_{if} - T_{An1} - \sum_{j=2}^{n} T_{Uj}}{T_{if}}. \tag{10}$$

Another aspect to take into account is the end node lifetime, supposing a limited (e.g., from a battery) power source. In particular, focusing on the power consumption and considering the worst case scenario in which the transaction is successfully acknowledged in the first RX1 window (so that no LoRaWAN retry is actually carried out), the LoRa-REP requires $n-1$ additional uplink messages, each one lasting $T_{Uj}j = 2 \cdots n$. The additional energy consumption (ΔE) can be estimated as in the following, starting from the consumption of the LoRa-REP approach (E) and the consumption of the regular LoRaWAN solution (E^L)

$$E = \left(\sum_{j=1}^{n} T_{Uj}\right) \cdot W_{TX} + T_{A11} \cdot W_{\mathrm{RX}} \tag{11}$$

$$E^L = T_{U1} \cdot W_{TX} + T_{A11} \cdot W_{\mathrm{RX}} \tag{12}$$

$$\Delta E = E - E^L = \left(\sum_{j=2}^{n} T_{Uj}\right) \cdot W_{TX} \tag{13}$$

where W_{TX} is the power consumption for transmitting and W_{RX} the power consumption for receiving. Therefore, the relative increment is

$$\frac{\Delta E}{E^L} = \frac{\left(\sum_{j=2}^{n} T_{Uj}\right) \cdot W_{TX}}{T_{U1} \cdot W_{TX} + T_{A11} \cdot W_{\mathrm{RX}}}. \tag{14}$$

However, considering that state-of-the-art single-chip transceiver draws tens of mA for receiving and transmitting messages lasting hundreds of ms, the absolute value of the additional energy ΔE required for a single LoRa-REP transaction is not significant, in the order of tens of mJ. Note that such a small ΔE value does not affect the overall device lifetime, since LoRa-REP usage is limited to emergency situations.

B. eNode: Versatile LoRaWAN Node

The feasibility of the proposed approach has been verified by means of a so-called eNode, where the "*e*" is for enhanced. Indeed, differently from regular nodes, the core of the eNode is a radio board hosting the SX1301-SX1257 chipset from Semtech, providing the baseband processor (the former) and digital-to-analog frontend (the latter). These devices are normally found in GWs (as the LoRa Gateway Board from Microchip we have actually used), and provide multiple digital receivers allowing to listen several channels and SFs simultaneously. However, in the developed eNode, a single digital transmitter is used. The radio board is managed by a Raspberry Pi 3, which is connected via an SPI serial link. The Raspberry implements upper protocol layers; in particular, the code is based on several modules. At the bottom, there is a hardware abstraction layer (HAL), derived from the "LoRa gateway" open-source software from "LoRa-Net" repository [27]. The HAL offers services to the "packet forwarder," based on Semtech open-source code as well.

It has to be noted that the original LoRa gateway software has been changed to invert the in-phase and quadrature components of the baseband signal of an incoming frame, allowing the eNode to behave as an end device capable of communicating with GWs. Additionally, the original "packet forwarder," i.e., the software responsible of injecting or retrieving wireless frames to or from the wired backhaul network, has been connected through a loopback interface to the software implementing the application-level services.

In particular, the "packet forwarder" software is in charge of managing a connectionless UDP/IP link for transferring JSON objects. Indeed, RxPK JSON objects are used to encapsulate downlink messages from the backend, while TxPK JSON objects encapsulate uplink ones from the field devices. All the JSON objects have a "tmst" item, which is the value of the internal SX1301 hardware time counter when the LoRa frames are received or sent, with microsecond granularity. Therefore, the tmst item can be used for precisely timestamping messages, taking into account that the timestamping point is the end of the frame for received ones, whereas the timestamping point is the beginning of the frame for transmitted ones.

In the eNode, the "packet forwarder" sends and receives the JSON objects via the loopback connection on the port 1680 to/from the application task, which is handled by Node-RED flows. In particular, the flow-based Node-RED development tool has been chosen since it natively supports the IoT paradigm. Node-RED allows for visual programming, providing a Web browser-based tool editor, which can be used to create Javascript function nodes, put into action by the underlying Node.js runtime.

Thanks to the decoupling between lower and upper sections of the protocol stack, the eNode can host several virtual nodes, all sharing the same hardware. For instance, in this work three different virtual nodes have been implemented, each one configured with the static SF value (in particular, the fastest data rates permitted by SF7, SF8, and SF9 have been considered).

According to the join procedure described by the LoRaWAN specifications, each virtual node has its own unique AppKey root key and DevEUI identifier. Differing from the regular LoRaWAN join procedure, all of them share a common JoinEUI identifier. On the backend side, there is no difference at all between virtual and real nodes, so that all of them must be configured individually as usual. A flow is executed only once at the beginning for handling the join procedure, in order to obtain the "nwkSKey" and "appSkey" session keys provided by the backend for authentication and enciphering, according to the so-called OTAA procedure (normally chosen in real deployments).

During regular data exchange, each virtual node is controlled by two separate flows: one devoted to uplink and the other to downlink management, as shown in Fig. 4. The function node "Get Keys VMote SFX" (where SFX stands for SF7, SF8, and SF9) returns the aforementioned "nwkSkey" and "appSkey" keys. The keys are then adopted into the "lora-packet-converter" function node, which manages LoRaWAN header parameters available as plain text, and takes care of the message ciphering and deciphering. The "TxPk Message" function node generates the uplink TxPk JSON object for the "packet forwarder." Finally, the "filter" function node is in charge of filtering the downlink messages, discarding all the messages that are not intended for the eNode.

Both the uplink and downlink related flows have timestamping function nodes, that enable the tracking of when the application generates the (uplink) message and the corresponding (downlink) acknowledgment is received, retrieving the local epoch time of the Raspberry Pi.

V. Experimental Reference Model

In this section, a short overview of the reference architecture considered for experimental evaluation of the proposed LoRa-REP is sketched out and shown in Fig. 5. In all the performed experiments, which took place in the same area with the same average interference, a single eNode is wirelessly connected to a regular LoRaWAN infrastructure implementing the NS and AS backend servers.

The full backward compatibility of the proposed solution has been demonstrated considering two different scenarios. In the first one, the eNode is connected to a public LoRaWAN backend infrastructure; the cloud-hosted Patavina Netsuite backend from A2A smart city has been used in this article. In the second one, a private, local backend, based on the open-source Chirpstack solution, has been considered.

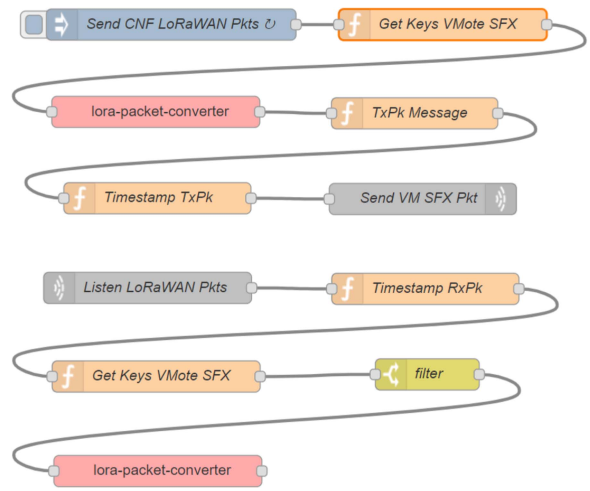

Fig. 4. Node-RED flows for the uplink (top) and downlink (bottom) of the virtual mote (confirmed LoRaWAN messages) inside the eNode architecture.

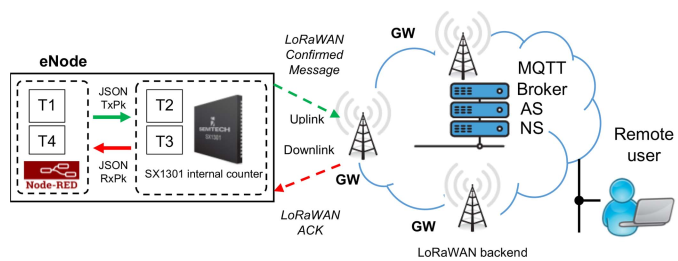

Fig. 5. Experimental testbed; the T1–T4 timestamps provided by the eNode are shown as well.

The proof-of-concept eNode device is configured to replicate application-level messages by means of three virtual nodes, VNY with $Y = 1 \cdots 3$, each one exploiting a fixed SF assignment (VN1 @ SF7, VN2 @ SF8, and VN3 @ SF9, respectively; in all cases, the coding rate is $\text{CR} = 4/5$). Transmission power is set at 14 dBm (i.e., 25 mW), which is the maximum value allowed in all the sub-bands of the unlicensed 868-MHZ band in Europe. Application-level messages are generated every 60 s and require acknowledgment (i.e., they are confirmed uplink messages). Each uplink message has a 40-B user data payload; when transmitted by a virtual node, the proper 13-B long LoRaWAN header is added. The acknowledge ACK, which is the downlink message sent by the NS, has a variable length, with a minimum raw payload of 12 B, possibly increased to 17 B if LoRaWAN MAC Commands are provided.

For sake of completeness, a specific setup for fairly comparing the LoRaWAN with retries and LoRa-REP has been added. Due to the limitations mentioned in Section II-C, the LoRaWAN with retries is emulated. The eNode is programmed to always transmit replicas (one for each virtual mote) of the same message using the sequence of SF derived from LoRaWAN specifications. Among all the possibilities, in the experiments of this article, the number of replicas is set to $Nb = 3$ and two sequences of SFs for the retries have been chosen in order to minimize the $D_{T,\max}$. In detail, the two sequences are: the SF7-SF7-SF7 (shortest delay) and the SF7-SF7-SF8 (longer delay but more robust). The transmission timing is equal to the LoRaWAN specifications. The outcome of every single transmission (one for each virtual mote) is recorded and, then, postprocessed. The overall transaction "fails" if all the three replicas do not receive their own acknowledge.

A. Performance Metrics

In order to assess the proposed approach performance, some metrics have been defined and evaluated. In this work, the focus is on the complete characterization of the LoRa-REP on the eNode side. Note that, conversely, evaluation of delays occurring from the on-field node to remote user has been already discussed in the literature; in general, hundreds of milliseconds are required for traversing the LoRaWAN infrastructure and deliver application data to the remote user [28].

In particular, the timestamps inside the Node-RED flow allow the estimation of delays at the application level. The delays at the data link level are evaluated using the baseband free running internal time counter. In details, four different timestamps TK_Y ($K = 1 \cdots 4$) have been collected per each individual transaction $Y = 1 \cdots 3$. Indeed, the transaction of the virtual node VNY starts at $T1_Y$, when the JSON TxPK confirmed uplink object is sent by the Raspberry (via UDP/IP) to the packet forwarder. Each of such LoRaWAN uplink message is actually transmitted by the SX1301 baseband processor at time $T2_Y$, according to its internal counter. The GWs receiving the uplink messages forward them to the backend, where authentication is performed by the NS and the ACK downlink is automatically scheduled in the following RX1 (or RX2) window. The LoRaWAN downlink message is received by the SX1301 at $T3_Y$ (always according to its internal counter). Next, a JSON RxPK is created and sent to the Raspberry in a UDP/IP packet. Finally, the Node-RED application receives the ACK at time $T4_Y$.

In accordance with the methodology described in [28], some metrics have been computed for single individual uplink–downlink transaction. The single transaction delay metric D_{TY} (i.e., the time needed to conclude a transaction) is estimated as

$$D_{TY} = T4_Y - T1_Y \tag{15}$$

the delay D_{IY} for traversing the LoRaWAN infrastructure (including the LoRaWAN backend servers and corresponding to the T_{macj} delay in Fig. 1) can be estimated as

$$D_{IY} = T3_Y - T2_Y \tag{16}$$

while the delay D_{AY} due to the application software (i.e., Node-RED in this article) can be estimated as

$$D_{AY} = D_{TY} - D_{IY} = (T4_Y + T2_Y) - (T1_Y + T3_Y). \tag{17}$$

Conversely, the overall transaction delay metric (i.e., the time needed to receive a confirmation from the communication partner) for the LoRa-REP is estimated as

$$D_T = T4_F - T1_1 \tag{18}$$

where $T4_F = \min(T4_1, T4_2, T4_3)$ is the timestamp of the first arrived downlink message in response to the LoRa-REP transaction started at $T1_1$. In other words, the LoRa-REP transaction starts using SF7 uplink and the useful downlink could be any one of the parallel transactions using SF7, SF8 or SF9. It has to be highlighted that synchronization between the Raspberry and the radio board is not required for computing the aforementioned metrics.

Finally, the effectiveness of the LoRa-REP for an emergency is evaluated in terms of probability of success of the emergency transaction, i.e., when at least one of the three possible downlink is correctly received

$$P_{\text{SUCCESS}} = \frac{\#\text{successful transactions}}{\#\text{transactions}} \tag{19}$$

trivially, the loss probability is evaluated as $P_{\text{LOSS}} = 1 - P_{\text{SUCCESS}}$. For the sake of comparison with regular LoRaWAN protocol applied to emergency message transmission, P_{SUCCESS} is also computed for any single uplink–downlink transaction and for the emulated LoRaWAN with the retry approach.

VI. Public Cloud Scenario

As previously stated, a public cloud-hosted backend has been considered for addressing typical smart city scenario. In particular, in this work, the public backend is provided by the "Brescia Smart Living" (BSL) project, financed by the Italian Ministry of Education, Universities and Research, permitted to develop and test IoT-based technologies for improving the efficiency, comfort and sustainability of the Brescia municipality. One of the project partners, A2A smart city, deployed a public LoRaWAN network covering the whole urban area (75 km^2 and 130 gateways). The backend is implemented by the Patavina Netsuite solution. The original target scenario included noncritical, nonreal-time applications, as monitoring the filling level of smart bins and dumpsters or monitoring the status of the district heater network. In such applications, tolerated update times range from few readings per day down to few readings per hour.

In this work, the authors verify the feasibility of the proposed LoRa-REP to further extend the range of applications, possibly, including emergency management. In particular, experiments have been carried out in both stationary and mobile scenarios. In the former case, the node is fixed and located inside one of the open space office at the Engineering Faculty of the University of Brescia. In the latter case, the node was hosted in a car moving in the city. In all cases, tests lasted two hours (i.e., 120 confirmed messages have been sent) and the ACK downlinks were always sent in the RX1 window.

A. Experimental Results With Fixed Node

In the experiment of this section, the eNode is fixed indoors, whereas the public gateway closest to the eNode is outdoors

TABLE I
PUBLIC CLOUD AND FIXED NODE SCENARIO: LORA RADIO CONDITIONS DURING EXPERIMENTS

	SF7		SF8		SF9	
	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)
Min	-9,5	-97,0	-10,0	-97,0	-9,8	-99,0
Ave	-6,3	-94,6	-5,9	-95,0	-5,9	-95,4
Max	-3,8	-93,0	-3,0	-92,0	-3,0	-92,0
St. dev	1,2	1,1	1,3	1,3	1,3	1,2

TABLE II
PUBLIC CLOUD AND FIXED NODE SCENARIO: THE SUCCESS PROBABILITY OF LORA-REP COMPARED WITH LORAWAN

		P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	29.2%	70.8%
	SF8 only	20.8%	79.2%
	SF9 only	15.8%	84.2%
LoRaWAN with retries	SF7-SF7-SF7	5.0%	95.0%
	SF7-SF7-SF8	2.5%	97.5%
LoRa-REP		6.7%	93.3%

TABLE III
PUBLIC CLOUD AND FIXED NODE SCENARIO: THE TRANSACTION DELAY D_T OF LORA-REP AND LORAWAN

D_T [ms]		min	ave	max	Stdev
LoRaWAN	SF7 only	1313.0	1323.8	1373.0	9.1
	SF8 only	1396.5	1421.4	1766.5	50.1
	SF9 only	1610.0	1635.6	1989.0	50.1
LoRaWAN with retries	SF7-SF7-SF7	1313.0	2804.6	11076.1	2559.9
	SF7-SF7-SF8	1313.0	2875.9	11151.1	2617.8
LoRa-REP		1313.0	1430.4	2047.0	215.0

TABLE IV
PUBLIC CLOUD AND FIXED NODE SCENARIO: THE INFRASTRUCTURE DELAY D_I AND APPLICATION DELAY D_A

Metric	SF	min	ave	max	Stdev
D_I [ms]	SF7	1149.0	1149.0	1149.0	0.0
	SF8	1257.0	1257.0	1257.0	0.0
	SF9	1473.1	1473.1	1473.1	0.0
D_A [ms]	SF7	164.0	174.8	224.0	9.0
	SF8	139.5	164.4	509.5	50.0
	SF9	136.9	162.6	515.9	50.3

(but within the University campus) at a distance of about 80 m. In order to assess the signal quality at the eNode side, the received signal strength indications (RSSI, in dBm), natively computed by the SX1301 baseband processor, is reported in Table I. It must be highlighted that sensitivity better than -140 dBm can be obtained with the adopted SX1257 front-end. Additionally, the SX1301 evaluated the signal-to-noise ratio (SNR, in dB) as well, which quantifies the actual capability to correctly decode an incoming frame (indeed, LoRa radios are able to decode packets having SNR smaller than −15 dB, e.g., as shown in [29]).

Thanks to the adoption of the LoRa-REP strategy, the success probability increases to 93.3% from the best, single SF transaction, which only offers 84.2% at SF9, as shown in Table II. When retry strategies are used, the original emergency packet loss of LoRaWAN is more than halved, at least. On the other hand, in this scenario, the LoRa-REP seems not offering advantages in terms of success probability over LoRaWAN with retries.

Time-related metrics are resumed in Tables III and IV. As expected, the overall transaction delay D_T of the LoRa-REP reaches a minimum value corresponding to the fastest, single SF transaction and allows to lower the average value with respect to the usage of SF8 and SF9 alone, respectively. The delay D_I for traversing the LoRaWAN infrastructure is in the order of 1.3 ms, and it depends on the actual message lengths. On the contrary, the delay D_A due to the application software depends on the tasks scheduling performed by the Node-RED environment, and has an average value smaller than 200 ms, despite some (few) outliers greater than 500 ms exist.

When LoRa-REP is compared with LoRaWAN with retries (see Table IV), the clear advantage of LoRa-REP emerges. With all the considered retry strategies, LoRaWAN maximum transaction delay is several times greater than the Lora-REP delay. Note that other LoRAWAN retry sequences (not used in this article) would increase the maximum delay even more.

B. Experimental Results With Mobile Node

In the experiment of this section, the eNode is moved along an outdoor (city wide) path. The track is chosen in order to have good radio coverage in all locations. The test itinerary is about 20-km long and was traveled by car; the maximum speed and average speed were 50 and 15 km/h. The number of reachable gateways ranges from one to eight. Extreme, average and standard deviation values of RSSI and SNR evaluated during the whole experiment are reported in Table V. It is evident that, despite minima being worse than in the fixed experiment (when the eNode, moving along the path, is more distant from the GWs), the average and maxima values are better (since some locations are closer to GWs). The slightly different radio coverage affects the success probability as well, which is slightly higher, as shown in Table VI. However, it is worth to note that in both cases, the LoRa-REP is able to increase the overall success probability close to 94%, confirming advantages of the proposed approach.

Also, the LoRaWAN with retries can increase the success probability, but less than LoRA-REP. The advantage of LoRa-REP in mobile scenario can be explained by the shadowing effect of the urban landscape than can block for several seconds the low SF transactions (with lower processing gain and sensitivity) used by LoRaWAN with retries. Additionally, mobile scenario is mainly outdoors, thus it is differently affected by multipath effect than fixed indoor scenario. In that situations LoRa-REP is still capable of successfully carry out the transaction exploiting the SF9 increased range and increased processing gain.

The overall transaction delay D_T is resumed in Table VII. As expected, there are no relevant differences with respect to the fixed scenario (indeed, infrastructure delay D_I and application delay D_A range in the same intervals as before and are not reported). Comparing LoRa-REP with LoRaWAN with retries gives indications similar to the fixed scenario: the large advantage of LoRa-REP is confirmed. Interesting to note, the average transaction delay for LoRaWAN with retries in mobile

TABLE V
PUBLIC CLOUD AND MOBILE NODE SCENARIO: LORA RADIO CONDITIONS DURING EXPERIMENTS

	SF7		SF8		SF9	
	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)
Min	-11,8	-120,0	-11,0	-120,0	-14,5	-120,0
Ave	2,7	-106,8	1,5	-110,0	-0,2	-111,7
Max	8,8	-71,0	8,8	-60,0	8,5	-72,0
St. dev	5,2	11,4	5,4	10,7	6,0	10,3

TABLE VI
PUBLIC CLOUD AND MOBILE NODE SCENARIO: THE SUCCESS PROBABILITY OF LORA-REP COMPARED WITH LORAWAN

		P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	20.3%	79.7%
	SF8 only	17.2%	82.8%
	SF9 only	13.3%	86.7%
LoRaWAN with retries	SF7-SF7-SF7	11.7%	88.3%
	SF7-SF7-SF8	10.9%	89.1%
LoRa-REP		6.2%	93.8%

TABLE VII
PUBLIC CLOUD AND MOBILE NODE SCENARIO: THE TRANSACTION DELAY D_T OF LORA-REP AND LORAWAN

D_T [ms]		min	ave	max	Stdev
LoRaWAN	SF7 only	1312.0	1326.2	1388.0	12.7
	SF8 only	1401.5	1421.2	2206.5	77.4
	SF9 only	1597.0	1625.7	1719.0	12.7
LoRaWAN with retries	SF7-SF7-SF7	1312.0	2127.8	10181.6	1870.6
	SF7-SF7-SF8	1312.0	2148.6	10709.6	1947.9
LoRa-REP		1312.0	1389.6	2371.0	180.7

scenario is lower than in the considered fixed scenario. This can be explained because in mobile scenario more transactions complete at the first attempt (i.e., they do not require a retry) with respect to the fixed scenario.

VII. PRIVATE LOCAL SCENARIO

The most likely scenarios for installation of a private LoRaWAN infrastructures are industrial premises and large, highly crowded, buildings with dense structure.

In order to deal with such situations, a specific experimental setup has been created at the University of Brescia Campus. It has been designed in order to highlight the feasibility and the performance of the proposed LoRa-REP in such harsh environments. In detail, the LoRaWAN network infrastructure is realized by a single Laird gateway (RG186) connected to a PC running the ChirpStack suite. ChirpStack (formerly known as "LoraServer") is a ready to use open-source implementation of complete LoRaWAN stack. The gateway is placed indoors at second floor (elevation 7 m) in the main building of the Engineering Campus.

All the experimental tests have been carried placing the eNode indoors in the fixed position at different elevations (mobility is typically not considered in such a scenario). Three different situations have been considered, in which the building structure presents several windows on both sides, but many armored concrete walls and floors can separate the gateway from the eNode during the experiments.

TABLE VIII
PRIVATE LOCAL SCENARIO: LORA RADIO CONDITIONS DURING EXPERIMENTS

	SF7		SF8		SF9	
	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)
Min	-11,8	-107,0	-11,5	-105,0	-12,0	-105,0
Ave	-1,3	-99,8	-1,4	-100,2	-1,6	-100,3
Max	5,8	-92,0	5,5	-91,0	5,8	-91,0
St. dev	3,4	2,5	3,6	2,3	4,0	2,5

TABLE IX
PRIVATE LOCAL SCENARIO: THE SUCCESS PROBABILITY OF LORA-REP COMPARED WITH LORAWAN

		P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	19.5%	80.5%
	SF8 only	11.5%	88.5%
	SF9 only	8.5%	91.5%
LoRaWAN with retries	SF7-SF7-SF7	7.5%	92.5%
	SF7-SF7-SF8	4.5%	95.5%
LoRa-REP		1.5%	98.5%

For the sake of comparison, the first experiment has been done in the same open space office at the Engineering Faculty used for the experiments with public infrastructures described in the previous section. The second experiment has been done in the industrial laboratory of the Engineering Faculty. This laboratory is located in another building, with a surface of about 2700 m^2, and typical heavy industry setup. The last experiment has been done in the University entrance hall during rush hour, simulating a heavy crowded place (like an airport). In all three experiments, the eNode was placed at 1 m elevation from the floor. The linear distances between the eNode and the gateway are: 1) 50 m for the open space experiment; 2) 70 m for the crowded hall experiment; and 3) 40 m for the industrial site experiment. In all cases, tests lasted 2 h (i.e., 120 confirmed messages have been sent) and the ACK downlinks are always sent in the RX1 window.

A. Experimental Results in the Open Space

The results obtained in this experiment can be directly compared with the results of the fixed node scenario of the public cloud scenario. Signal quality for both uplink and downlink messages is reported in Table VIII. A comparison with corresponding values reported in Table I highlights slightly better conditions, due to a better (closer) placement of the private gateway.

The LoRa-REP approach easily gains over the use of single SF with an overall success probability of 98.5%. In other terms, it means a reduction to about one-sixth of the original emergency packet loss when the best performing, single SF, the transaction is addressed (i.e., from 8.5% using SF9 only to 1.5% using LoRa-REP), as shown in Table IX. Note that the better positioning of the private gateway permits performance even better than those reported in Table II. Finally, in this scenario, the LoRa-REP has a small advantage over LoRaWAN with retries, probably due to the use of SF9 for one of the three replicas. The LoRa-REP protocol is also able to deliver the messages with an average overall delay that is lower than

TABLE X
PRIVATE LOCAL SCENARIO: THE TRANSACTION DELAY D_T OF LORA-REP AND LORAWAN

D_T [ms]		min	ave	max	Stdev
LoRaWAN	SF7 only	1310.0	1321.4	1391.0	10.0
	SF8 only	1406.0	1425.5	1825.5	39.1
	SF9 only	1597.0	1624.8	1950.0	38.3
LoRaWAN with retries	SF7-SF7-SF7	1310.0	2236.4	10314.8	2138.1
	SF7-SF7-SF8	1310.0	2374.6	10978.8	2411.2
LoRa-REP		1310.0	1406.6	2360.0	206.9

TABLE XI
PRIVATE LOCAL SCENARIO: THE INFRASTRUCTURE DELAY D_I AND APPLICATION DELAY D_A

Metric	SF	min	ave	max	Stdev
D_I [ms]	SF7	1143.0	1143.0	1143.0	0.0
	SF8	1267.2	1267.2	1267.2	0.0
	SF9	1473.0	1473.1	1473.1	0.0
D_A [ms]	SF7	166.2	177.5	247.2	10.03
	SF8	139.3	158.3	558.3	39.1
	SF9	123.9	151.7	476.9	38.2

TABLE XII
PRIVATE LOCAL SCENARIO: THE SUCCESS PROBABILITY OF LORA-REP COMPARED WITH LORAWAN IN DIFFERENT SCENARIOS

		Crowded Hall		Industrial building	
		P_{LOSS}	$P_{SUCCESS}$	P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	16.5%	83.5%	2%	98%
	SF8 only	7.5%	93.5%	1%	99%
	SF9 only	4.5%	95.5%	0.5%	99.5%
LoRa-REP		2.5%	97.5%	0.5%	99.5%

the most robust single SF (SF9), as visible in Table X. The great advantage of LoRa-REP over LoRaWAN with retries in terms of delay is also shown in Table X. Finally, as expected the application delay and infrastructure delay, shown in Table XI, are pretty similar to the ones of the public scenarios, because the Node-Red application and the eNode hardware are the same.

B. Other Experiments and Discussion

The results obtained in the last two experiments in terms of success/loss probabilities are reported in Table XII. Again the LoRa-REP was able to deliver more emergency packets than using a single SF. The gain, in this case, was of about two times for the crowded hall, while it is very low for the industrial building. The reason of the high success ratio of single SF in the industrial building is explained by: huge free space inside this site (there are no partition walls); the reduced linear distance between the eNode and the gateway; and the presence of large windows (easily traversed by the wireless signal) on the sides of both main building and industrial laboratory building.

Overall, the results obtained in the private local scenario further confirm the advantages of the proposed LoRa-REP protocol, when it is applied transparently to different LoRaWAN infrastructure. As a matter of fact, the underlying LoRaWAN backend (cloud or local) does not affect the LoRa-REP behavior, provided that it is able to process incoming confirmed uplink within the LoRaWAN downlink windows RX1 or RX2.

VIII. CONCLUSION

LPWANs have emerged as a viable solution to address many of the challenges posed by IoT-like applications. In particular, LoRaWAN is known for its openness and flexibility. In this article, we propose and discuss characteristics and the performance of LoRa-REP, a novel replication scheme which allows to step up the success probability in completing confirmed data exchanges, as required, for instance, in emergency management.

The vast measurement campaign, carried out in different, real-world scenarios, highlighted the compatibility with already in place deployments in smart city, industrial, and building automation applications, thus preserving investments. The results of the real use cases clearly prove that the proposed approach is effective in reducing the loss of emergency messages of at least two times (and up to 13 times) with respect to LoRaWAN without retries. Moreover, the average transaction time is reduced by 200 ms in the worst use cases. On the other hand, if LoRa-REP is compared with LoRaWAN with retries, its advantages are: a higher success probability in case of scenarios with highly (and rapidly) changing conditions (e.g., mobile); a more general reduction of the average delay (up to 1400 ms in the worst use cases); and a dramatic reduction of the maximum delay (e.g., five times less).

REFERENCES

[1] M. Compare, P. Baraldi, and E. Zio, "Challenges to IoT-enabled predictive maintenance for industry 4.0," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4585–4597, May 2020.

[2] S. H. Chae, C. Jeong, and S. H. Lim, "Simultaneous wireless information and power transfer for Internet of Things sensor networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2829–2843, Aug. 2018.

[3] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic, "High-reliability and low-latency wireless communication for Internet of Things: Challenges, fundamentals, and enabling technologies," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7946–7970, Oct. 2019.

[4] M. Rady, M. Hafeez, and S. A. R. Zaidi, "Computational methods for network-aware and network-agnostic IoT low power wide area networks (LPWANs)," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5732–5744, Jun. 2019.

[5] M. Pasetti, P. Ferrari, D. R. C. Silva, I. Silva, and E. Sisinni, "On the use of LoRaWAN for the monitoring and control of distributed energy resources in a smart campus," *Appl. Sci. (Switzerland)*, vol. 10, no. 1, p. 320, 2020

[6] M. Vilgelm, S. R. Linares, and W. Kellerer, "Dynamic binary countdown for massive IoT random access in dense 5G networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6896–6908, Aug. 2019.

[7] *AN1200.22 LoRa™ Modulation Basics*, Semtech Corp., Camarillo, CA, USA, May 2015. [Online]. Available: https://lora-developers.semtech.com/

[8] E. Sisinni *et al.*, "LoRaWAN range extender for industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5607–5616, Aug. 2020.

[9] A. Hoeller, R. D. Souza, O. L. Alcaraz Lòpez, H. Alves, M. de Noronha Neto, and G. Brante, "Exploiting time diversity of LoRa networks through optimum message replication," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Lisbon, Portugal, 2018, pp. 1–5.

[10] J. M. D. S. Sant'Ana, A. Hoeller, R. D. Souza, S. Montejo-Sánchez, H. Alves, and M. D. Noronha-Neto, "Hybrid coded replication in LoRa networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5577–5585, Aug. 2020.

[11] M. Centenaro, L. Vangelista, and R. Kohno, "On the impact of downlink feedback on LoRa performance," in *Proc. IEEE 28th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, 2017, pp. 1–6.

[12] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on the viability of confirmed traffic in a LoRaWAN," *IEEE Access*, vol. 8, pp. 9296–9311, 2020.
[13] A. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?" in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, 2017, pp. 1–6.
[14] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of IoT traffic and its impact on LoRaWAN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, 2017, pp. 1–6.
[15] A. Burns, R. I. Davis, S. Baruah, and I. Bate, "Robust mixed-criticality systems," *IEEE Trans. Comput.*, vol. 67, no. 10, pp. 1478–1491, Oct. 2018.
[16] H. Farag, E. Sisinni, M. Gidlund, and P. Österberg, "Priority-aware wireless fieldbus protocol for mixed-criticality industrial wireless sensor networks," *IEEE Sensors J.*, vol. 19, no. 7, pp. 2767–2780, Apr. 2019.
[17] J.-S. Fu, Y. Liu, H.-C. Chao, and Z.-J. Zhang, "Green alarm systems driven by emergencies in industrial wireless sensor networks," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 16–21, Oct. 2016.
[18] A. Willig, "Redundancy concepts to increase transmission reliability in wireless industrial LANs," *IEEE Trans. Ind. Informat.*, vol. 1, no. 3, pp. 173–182, Aug. 2005.
[19] G. Cena, S. Scanzio, and A. Valenzano, "Experimental evaluation of seamless redundancy applied to industrial Wi-Fi networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 856–865, Apr. 2017.
[20] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 824–835, Feb. 2014.
[21] L. Leonardi, F. Battaglia, and L. Lo Bello, "RT-LoRa: A medium access strategy to support real-time flows over LoRa-based networks for industrial IoT applications," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10812–10823, Dec. 2019.
[22] R. Abhishek, S. Zhao, D. Tipper, and D. Medhi, "SeSAMe: Software defined smart home alert management system for smart communities," in *Proc. IEEE Int. Symp. Local Metropolitan Area Netw.*, Osaka, Japan, Jun. 2017, pp. 1–6.
[23] F. R. Bhatti, M. A. S. Shah, C. Maple, and S. Ul Islam, "A novel Internet of Things-enabled accident detection and reporting system for smart city environments," *Sensors (Switzerland)*, vol. 19, no. 9, p. 2071, 2019.
[24] S. Borkotoky, C. Bettstetter, U. Schilcher, and C. Raffelsberger, "Allocation of repetition redundancy in LoRa," in *Proc. 25th Eur. Wireless Conf.*, Aarhus, Denmark, 2019, pp. 1–6.
[25] P. Marcelis, V. Rao, and R. V. Prasad, "DaRe: Data recovery through application layer coding for LoRaWAN," in *Proc. ACM/IEEE 2nd Int. Conf. Internet Things Design Implement.*, Pittsburgh, PA, USA, Apr. 2017, pp. 97–108.
[26] M. Luvisotto, F. Tramarin, L. Vangelista, and S. Vitturi, "On the use of LoRaWAN for indoor industrial IoT applications," *Wireless Commun. Mobile Comput.*, vol. 2018, May 2018, Art. no. 3982646, doi: 10.1155/2018/3982646.
[27] *LoRa*. Accessed: Apr. 2, 2020. [Online]. Available: https://github.com/Lora-net/
[28] D. F. Carvalho *et al.*, "A test methodology for evaluating architectural delays of LoRaWAn implementations," *Pervasive Mobile Comput.*, vol. 56, no. 5, pp. 1–17, 2019.
[29] P. Neumann, J. Montavont, and T. Noël, "Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study," in *Proc. IEEE 12th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, New York, NY, USA, 2016, pp. 1–8.

Emiliano Sisinni (Member, IEEE) received the M.Sc. degree in electronics engineering and the Ph.D. degree in electronic instrumentation from the University of Brescia, Brescia, Italy, in 2000 and 2004, respectively.

He is currently a Full Professor with the Department of Information Engineering, University of Brescia. His research interests include wireless and wired networking for IIoT applications.

Dhiego Fernandes Carvalho (Graduate Student Member, IEEE) received the B.Sc. degree in computer engineering and the M.Sc. degree in computer networking from the Federal University of Rio Grande do Norte, Natal, Brazil, in 2008 and 2014, respectively. He is currently pursuing the Ph.D. degree in technology for health course.

his research interests are focused on low-power wide area networks for smart cities and smart applications for industry 4.0.

Paolo Ferrari (Member, IEEE) received the M.Sc. degree in electronics engineering and the Ph.D. degree in electronic instrumentation from the University of Brescia, Brescia, Italy, in 1999 and 2003, respectively.

He is currently a Full Professor with the Department of Information Engineering, University of Brescia. His research is about measurement systems for industrial applications, including performance and security analysis of real-time networks and IoT applications.