IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received December 13, 2019, accepted December 30, 2019, date of publication January 8, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964909

A Survey on the Viability of Confirmed Traffic in a LoRaWAN

JACO MORNÉ MARAIS[1], (Student Member, IEEE),
ADNAN M. ABU-MAHFOUZ[1,2], (Senior Member, IEEE),
AND GERHARD P. HANCKE[1], (Life Fellow, IEEE)
[1]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 00028, South Africa
[2]Council for Scientific and Industrial Research, Pretoria 0083, South Africa

Corresponding author: Jaco Morné Marais (u11024063@tuks.co.za)

This work was supported by the Council for Scientific and Industrial Research of South Africa and Telkom.

ABSTRACT Internet of Things (IoT) deployments are on the rise globally with Low Power Wide Area Networks (LPWAN) providing the wireless networks needed for this expansion. One of these technologies namely Long Range Wide Area Network (LoRaWAN) has proven to be a very popular choice. The LoRaWAN protocol allows for confirmed traffic from the end device to the gateway (uplink) and the reverse (downlink), increasing the number of IoT use cases that it can support. However, this comes at a cost as downlink traffic severely impacts scalability due to in part a gateway's duty cycle restrictions. This paper highlights some of the use cases that require confirmed traffic, examines the recent works focused on LoRaWAN confirmed traffic and discusses the mechanism with which is implemented. It was found that confirmed traffic is viable in small networks, especially when data transfer is infrequent. Additionally, the following aspects negatively impact the viability of confirmed traffic in large networks: the duty cycle restrictions placed on gateways, the use of spreading factor 12 for receive window 2 transmissions, a high maximum number of transmissions (NbTrans) and the ACK_TIMEOUT transmission backoff interval. The paper also raises and suggests solutions to open research challenges that must be overcome to increase the viability of confirmed traffic.

INDEX TERMS LoRaWAN, LPWAN, IoT, ACK.

I. INTRODUCTION

The Internet of Things (IoT) industry is growing rapidly with more and more devices being deployed. These devices are frequently battery-powered and are scattered over large areas, thus requiring communication technologies designed to meet their needs [1]. This need has been met through the development of Low Power Wide Area Network (LPWAN) technologies such as Sigfox, NB-IoT and the focus of this paper: LoRa Wireless Area Network (LoRaWAN).

Since the release of the LoRaWAN protocol in 2015, the conducted research has focused on a few areas such as exploring the Physical layer (PHY), performance evaluations examining range and throughput, and the protocol's Adaptive Data Rate (ADR) scheme. Surveys on the LoRaWAN protocol, such as [1]–[4] all examined the published LoRaWAN literature and attempted to classify the work into sections. These works currently cover multiple topics such as security, communication range, energy consumption, link coordination and suitable LoRaWAN applications. These surveys are very valuable, but as they are not focused on a specific topic, do not explore an aspect such as ACKs extensively.

The negative impact of ACKs on network performance is noted in [2], and the authors suggest the creation of a dynamic retransmission policy. They further note that the factors triggering retransmission must be extensively studied before a dynamic policy is created. The review presented in [1], examines several works studying performance at the physical and network levels. The review examines several topics such as interference (from other technologies and self-interference), power consumption, security and the ADR scheme but only briefly discusses acknowledgements.

When the literature was surveyed in [3], it was found that most works focused on coverage tests and the various applications LoRaWAN could be suited for. The coverage of

The associate editor coordinating the review of this manuscript and approving it for publication was Tawfik Al-Hadhrami.

acknowledgements, however, are limited to their description as part of the section detailing the LoRaWAN protocol's inner workings. A total of 54 papers were classified into 5 categories in [4]. It was noted that the current trend is performance evaluations through experimental work and a need for simulation and modelling work was identified. The role of acknowledgements was not an aspect investigated in this paper.

The protocol could be the technology needed for IoT deployments to reach proper scale, but several IoT use cases are only viable if their required Quality of Service (QoS) levels can be met [5]. LPWAN technologies are frequently marketed as being able to support very large networks, but these claims have some conditions which are only revealed once you examine the QoS levels achievable in a large network [6].

The QoS needs in a Wireless Sensor Network (WSN) tends to differ from those in wired networks as the limited device resources, the large number of nodes, different traffic types and unreliability of radio transmissions must be dealt with [7]. IoT applications are also unique and depending on the IoT use case, different QoS parameters such as latency, data rate or reliability are key [5].

This work focuses on one aspect of the LoRaWAN protocol that underpins its QoS capabilities: the use of acknowledgements. The authors of [8] showed how packet delivery was improved significantly once retransmissions were enabled in a small testbed. However, the results obtained there are not translatable to the performance of a large LoRaWAN with thousands of devices. Additionally, acknowledgements are not only used to confirm application layer packets, but several MAC commands also requires confirmation of reception and execution [9]. A detailed evaluation of the viability of confirmed traffic in LoRaWANs is required to better understand the QoS levels achievable with this technology.

This paper contributes the following.

- It discusses the IoT use cases that require confirmed traffic.
- It provides an extensive survey of all the LoRaWAN literature that deals with confirmed traffic and discusses some identified common themes.
- It presents the challenges that hamper the viability of confirmed traffic and proposes methods to counteract them.

The rest of this paper is organised as follows. Section II discusses the various IoT use cases that require confirmed traffic. Background information on the LoRaWAN protocol as well the mechanisms enabling confirmed traffic is presented in Section III. An extensive survey on the impact of downlink traffic, with a focus on confirmed traffic, can be found in Section IV. A discussion on the surveyed work is located in Section V. Alternative methods to implement confirmed traffic is discussed in Section VI. In Section VII, the open challenges impacting the viability of confirmed traffic are discussed. Finally, the work is concluded in Section VIII.

II. USE CASES THAT REQUIRE ACKS

At a glance, most IoT applications appear similar and can be considered as a large number of battery-powered devices who infrequently send small amounts of data. The assumption is made that the value lies in collecting a large amount of data and so the delivery of specific packets becomes less important than collecting sufficient data. However, when IoT use cases are compared with one other, their unique QoS requirements start to be revealed [10].

The QoS requirements for a moisture measuring device part of a smart agriculture IoT project are minimal. The device's task is simply to periodically deliver a moisture reading, of which a few can get lost without causing any problems. For an IoT smoke detector, however, a confirmation that its alert has been received is critical.

There are potentially multiple QoS metrics that must be met when IoT use cases are examined, this paper focuses on one QoS criteria namely reliability and how this can be achieved through the use of acknowledgements. As pointed out in [10], focusing on one QoS metric frequently comes at the cost of other metrics. In their network simulations, it was found that maximum capacity was achieved only at very low packet success rates as there exist tradeoffs between the metrics. There are several use cases in which the confirmation that a data packet or a command has been received is critical. These have been classified in a few broad categories namely Smart city, Smart health, Industrial and Critical situations. This is not meant to be an extensive list, but rather to highlight that there are use cases in these popular IoT deployment domains that require confirmed traffic.

A. SMART CITY

An area in which LoRaWAN deployments are highly suitable is smart city operations, as these involve a high amount of devices that transmit only a few times a day [6], [11]. A smart city's network would include several thousands of sensors monitoring aspects of the city such as air and water quality, road conditions and traffic congestion [12]. These aspects could all make use of unconfirmed traffic, but others such as smart lighting and smart metering require acknowledgements for commands transmitted to the devices.

Smart metering involves the remote monitoring and control of electricity, water or gas consumption and would involve thousands of devices in urban cities [13]. Smart meters are part of the large resource networks responsible for keeping the city functional. Worldwide, the electricity sector has increasingly been aiming to optimise and automate the generation, transmission and distribution of electricity through the creation of smart grids, with smart meters being involved in this process [14]. LoRaWAN is a strong contender to enable the installation of smart meters due to the urban ranges possible and the number of devices a single gateway can support [14]. Acknowledgements play an important part in allowing providers to use their smart meters to e.g. switch off a non-paying customer's electricity.

LoRa's low power consumption makes it an attractive choice for smart city deployments as frequent battery changing would be difficult in several applications. The technology has also shown the potential to be combined with energy harvesters to e.g. not only monitor traffic flow but be powered from the vibrations generated by vehicles [15].

B. SMART HEALTH

Smart health is an area in which tremendous growth is expected to occur due to technological advancements and the rise in the global life expectancy [16]. This area is quite broad as it contains not only the detection of an illness but also the long term monitoring of health conditions/illnesses. Smart health applications frequently have strict QoS requirements, but this is also application-specific. Real-time systems monitoring glucose levels or heart rhythms require low communication delay, whilst other systems such as fall detection are sensitive to packet loss [5], [16]. These systems differ as some systems simply transmit sensor data whilst others process the data and only send updates.

One example of a system where the data is processed locally and only alerts are sent is fall detection [17]. A device is attached to a person and continually examines accelerometer data to determine when an alert to indicate a fall has occurred must be sent. Another example is the use of LoRaWAN in an IoT bio-fluid analyzer testing for urinary tract infections [18]. The analyzer sends the result of a test to a remote secure server, the use of acknowledgements here can ensure that the patient's test results were captured.

LoRaWAN was evaluated as a potential technology for smart health in [19]. It was deemed to be suited in most cases due to its range, latency and network capacity capabilities. The use of Class A was deemed a good compromise between battery efficiency and receiving an acknowledgement for transmitting health data. In many cases it is not that all data be received successfully, only critical data such as an alert of a potential heart attack or fall will require acknowledgement.

Another aspect of smart health is ensuring the cold chain is kept intact for products such as insulin and vaccines [20]. These products have strict acceptable temperatures ranges in which items must be kept during their transportation, storage and handling. If this acceptable range is breached, it is critical that any sent alerts are received, as this item should no longer be considered safe to use.

C. INDUSTRIAL

One of the major areas in which the IoT is expected to play a major role is in industrial settings, hence the term Industrial Internet of Things (IIoT) was coined. The deployment of low-cost connected devices is seen to pave the way forward to productivity improvements and cost savings across manufacturing and supply chains [21]. Industrial settings can have harsher requirements than other IoT deployments areas and deployment must be economically viable [16].

The use of LoRaWAN in an industrial setting is up for debate as whilst the technology offers good scalability, range and security it might not meet a specific use case's jitter, delay or bandwidth requirements [22]. The protocol was not designed with delay in mind, and even for small 10 byte long packets sent using the fastest LoRaWAN data rate available, the transmission time required would be 20 ms. This delay would be too high for real-time operations such as closed-loop control but suitable for most other operations [6], [16].

For industrial settings with real-time requirements, alternatives to the use of LoRaWAN as the MAC layer have been developed. Examples of this are RT-LoRa ([23]) and Industrial LoRa ([24]) which aims to optimise message latency.

Industrial settings vary greatly from crowded factories in cities with lots of radio interference to wide-open areas such as open-pit mining. The number of devices would also differ from thousands of devices down to potentially only one hundred. The network might also be split between a majority of devices that periodically sent measurements and a few devices whose packets are critical to be received [16]. In such a scenario, it would be beneficial to only deploy one network servicing both needs rather than deploying two networks.

In industrial settings, the fact that the protocol supports confirmed traffic is especially useful as LoRaWAN devices would commonly be used to monitor equipment [25]. During normal operations, non-critical data such as temperature would be transmitted for use in fault prevention strategies. Critical faults, such as a breakdown or measured values outside of acceptable operating ranges can be sent as confirmed traffic, to make sure a manufacturing line can be halted.

A LoRaWAN supports confirmed uplink and downlink traffic and its viability in industrial settings will in part depend on the power sources available. Battery-powered devices will necessitate the use of LoRaWAN Class A, whilst devices with grid power can use the power-intensive but lower delay Class C.

An industry in which the IoT can play a significant role is Smart agriculture. This sector traditionally had limited monitoring and automation options but as land and water become scarce better ways to farm as needed [21].

Animal tracking is a use case where LoRaWAN is starting to feature in strongly, with commercial applications such as mOOvement's cattle tracker already available [26]. The technology is highly suited for this type of use case due to its excellent coverage capabilities in rural areas [10]. Most of the animal's movement can be sent without requiring acknowledgements, however, once an animal breach geo-fences due to e.g. theft, confirmation of the animal's location becomes important.

Similarly, when technology is used to enable precision agriculture, most of the sensor measurements can be sent as unconfirmed messages. Only the devices that enable the automation elements on a smart farm, such as irrigation pumps, need to make use of acknowledgements to ensure proper operation.

D. CRITICAL SITUATIONS

One area in which the IoT can be of great benefit is during emergency scenarios or natural disasters. Due to its scaling and long-range capabilities, LoRaWAN has started to attract the attention of researchers building networks for these critical situations [27]. IoT deployments in these situations will have to share the limited available resources whilst ensuring QoS requirements are met.

For the reaction to an event to be effective, information about the event and the environment in which it occurs is key [28]. As the situation unfolds, data from different nodes will become more critical than others, and here the use of priority based acknowledgements can be put to use.

The emergency services could also make use of a LoRaWAN to enable basic communication [29]. During disasters, traditional networks are frequently inactive or overwhelmed. Whilst a LoRaWAN does not have the bandwidth capabilities of these networks, it can be used to sent text information and can provide a reliable network.

E. COMMON REQUIREMENTS

All of the use cases mentioned previously require that the network supports downlink traffic. The downlink traffic volume is much lower than the uplink volume but still plays a vital part in making the network viable. Other use cases normally considered to be uplink only such as collecting sensor readings can still benefit from downlink traffic capability. The use of groupcasting and geocasting, [30], can be an effective way of reducing traffic by specifying which nodes should transmit data whilst the rest can remain in sleep modes. In this manner, downlink traffic can be used to improve network performance and only the data that has high value will be transmitted and stored.

A common theme across all of these use cases is the need for not only acknowledgements but that traffic should be spliced so that critical applications can be assured of delivery. One method, suggested in [6], is that new channel hopping methods are used with dedicated uplink and downlink channels for certain (critical) packets.

III. THE LoRaWAN PROTOCOL

Unlike some of its competitors, the LoRaWAN standard is an open standard and was developed by the LoRa Alliance. Due to its openness, it has proven to be very popular amongst academics, industry and the maker community. The PHY of a LoRaWAN network is a modulation technique called Long Range (LoRa) and was developed by Cycleo before being acquired by Semtech and remains proprietary. LoRaWANs operate in the Industrial, Scientific and Medical (ISM) bands and its operation thus differs depending on the network's region. Note that this paper was written from the perspective of EU 868 networks with the regional parameters specified in [31].

A LoRaWAN compliant radio offers several data rates and packets have maximum packet lengths dependent on the data rate used for their transmission. When the data rate is set, either statically or dynamically through LoRaWAN's ADR scheme, this is translated into two other settings namely Spreading Factor (SF) and Bandwidth (BW). In the European Union (EU) region, a LoRaWAN compliant radio has two BW settings namely 125 kHz and 250 kHz and offers six different orthogonal SF values namely 7-12. The choice of SF involves a trade-off between the achievable distance and throughput, higher SFs offer more range but also have higher time-on-air durations causing Duty Cycle (DC) restrictions to be reached earlier [13]. These DC restrictions are region-specific and must be adhered to in LoRaWAN networks.

The LoRaWAN standard adds the required Medium Access Control (MAC) functionality and specifies that the network has a star-of-stars topology which supports uplink and downlink messages. The network is managed by a central Network Server (NS) which is responsible for managing packet duplicates, scheduling acknowledgements, the ADR scheme and will reroute any packets to the Application Server they are destined for [32], [33].

Uplink messages are sent by a device to the NS and will be received by one or more gateways who all forward the packet to the NS. Downlink messages are sent from the NS to a specific device and will be transmitted by only one gateway chosen by the NS. LoRaWAN gateways are based on Semtech's SX1301 chip, capable of demodulating 8 LoRaWAN frames simultaneously [34]. LoRaWAN gateways are half-duplex and thus a gateway is unable to receive any transmissions whilst it is transmitting downlink messages.

Since the release of version 1.0.0 in January 2015, the LoRaWAN specification has continually been improved. The latest minor update namely revision 1.0.3 was released in July 2018. A major update namely 1.1.0 was released in October 2017 and is backwards compatible with 1.0.x end devices and networks. The updates have, amongst other things, added new MAC commands and deprecated old ones, increased security and added frequency plans.

The protocol specifies three possible devices classes all allowing bi-directional communication and are referred to as class A, B or C. With class A each uplink from a device to the gateway is followed by two short downlink receive windows as shown in Figure 1. The device can only receive communication from the gateway during these two windows.

This optimises power consumption but causes devices to be unreachable until they sent a message. Class B adds additional scheduled receive windows to make devices more reachable and downlink communication frequency more predictable. This is done through the gateway transmitting a time synchronised beacon and as devices must remain synchronised, increases a device's power consumption. The final class, C, is best used by devices without power concerns as these devices have near-continuous receive windows. This allows downlink communication to class C devices to have lower latency than the other classes but has no impact on uplink latency.

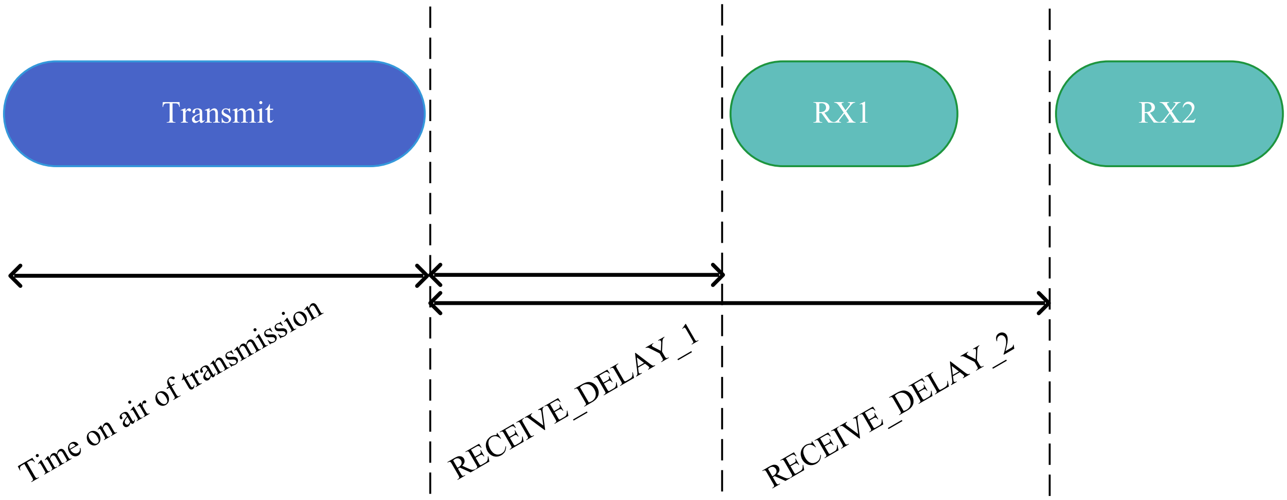

FIGURE 1. Timing diagram for receive windows for Class A devices. Adapted from [33].

The most commonly deployed class in a LoRaWAN is envisioned to be class A. In terms of receiving acknowledgements, this class has the strictest requirements, as class A devices will only be able to receive an Acknowledgement (ACK) during their two downlink receive windows. Whenever an ACK is requested from the device, as required as a response to several MAC commands, the device should transmit an ACK as soon as possible [9], [33]. The device may transmit immediately a potentially empty packet with the ACK bit set or simply set the ACK bit of its next message.

The protocol specifies three default channels in the EU 868.0-868.6 MHz h1.5 sub-band that all devices on the EU 868 channel plan must support. For these channels, the maximum transmit power is +14 dBi Equivalent Isotropically Radiated Power (EIRP) and a sub-band DC limitation of < 1 % must be adhered to. By default, a device's second receive window is configured to transmit on 869.525 MHz using the maximum spreading factor (SF12) with sub-band h1.7 allowing a DC of 10 % [35], [36]. More channels can be added depending on the region, for example, the popular The Things Network (TTN) uses 10 channels in its EU 863-870 channel plan [37].

Figure 1 also shows the timing used in the opening of the receive windows after a device's uplink transmission. The first window opens after the uplink's modulation finishes (exact time specified by RECEIVE_DELAY1 $+/-$ 20 μs) and will by default use the same channel and data rate as the uplink. The second window opens RECEIVE_DELAY2 seconds after the uplink and its frequency and data rate are configurable and region-specific [31], [33]. Transmitting an ACK using the first window could result in a lower transmission time as transmitting an ACK on any of the lower SFs would be faster than doing so on SF12. The first window may, however, not be available as it will have a tighter DC restriction than the second window. The default timing values for these delays as well as the delay between retransmission attempts when an ACK was not received is given in Table 1.

TABLE 1. Default parameter settings in EU 868 networks.

Parameter	Default setting
RECEIVE_DELAY1	1 second.
RECEIVE_DELAY2	RECEIVE_DELAY1 + 1 second.
ACK_TIMEOUT	random delay between 1 and 3 seconds.

A. LoRaWAN PACKET STRUCTURE

LoRaWAN packets consist of a Preamble used to synchronise the receiver, an optional header and its Cyclic Redundancy Check (CRC), the payload and then an optional payload CRC [33]. The CRC is always excluded in downlink packets to keep their size to a minimum. LoRaWAN packets also have two format options namely explicit and implicit of which explicit is the default [38]. The explicit mode adds the header mentioned above which specifies the payload length (in bytes), the coding rate and if a CRC is provided for the payload. When the implicit mode is used, this information is excluded thereby reducing transmission time and manually configured on either side.

The LoRaWAN protocol supports unconfirmed uplink and downlink transmissions as well as a confirmed version of both. By default, all traffic is sent as unconfirmed messages. The message type is specified by the MType bit field in the MAC header part of the packet and the protocol allows a single packet to contain application data and MAC commands.

The MAC payload of a data message contains a frame header (FHDR) and acknowledgements are managed by the frame control octet (FCtrl) part of this header. It is through this octet that a device can request an ACK or an ADR acknowledgement request, a crucial part of the ADR scheme. As the ADR scheme aims to optimise a device's data rate and transmit power, it could potentially dictate settings that result in a loss of network connectivity. As most data would be sent as unconfirmed messages, the device needs a method of confirming that the gateway is receiving the sent data. Should the ADR scheme be used, the protocol requires that the end device keeps track of how many messages it has sent since it last received a downlink transmission. Once this exceeds a threshold, the ADRACKReq bit is set in the next message. Upon detecting this bit, the gateway must transmit a downlink message thereby confirming that the device is still connected. If no such confirmation is received, the device proceeds to adjust its data rate and transmit power until it is reconnected.

B. PROCEDURE WHEN AN ACK IS NOT RECEIVED

There are several causes for failure to receive an ACK: the confirmed message was not received by the other party, DC limitations prevented the transmission of the ACK and finally, the ACK was transmitted but not received. Another potential cause could be the delay between the NS receiving the request, selecting a gateway from which to acknowledge and the chosen gateway transmitting the ACK.

The retransmission procedure when an ACK is not received is one part of the LoRaWAN protocol where version 1.1 brings significant changes compared to previous versions. LoRaWAN versions can be split between 1.0.x, where x equals 0, 1, 2 and 3 and the new versions 1.1.x for which x is currently 0. In version 1.0.3 [39], much of this procedure is left to the developer of the end device and NS. The number of retransmissions, when they occur and what to do after these retransmissions are still unsuccessful are not specified. A recommended retransmission strategy is however provided in Chapter 18 of the protocol ([39]). Additionally, a retransmission back-off strategy is provided in Chapter 7 for retransmissions triggered by external events causing synchronization between devices ([39].

This strategy spaces retransmissions ACK_TIMEOUT (a random value between 1 and 3) seconds after the second receive window, opts for a different channel than was used for the uplink and lowers the data rate of retransmission attempts. It is left to the application and NS developers to decide what should happen if transmission remains unsuccessful: either continue attempting to send the same data or simply discard it and send the next message.

Version 1.1, [33], differentiates the response to either downlink and uplink frames and also specifies that the NbTrans field's value now also applies to confirmed uplink frames and no longer to only unconfirmed frames. Additionally, version 1.1 does not recommend changing the data rate for retransmission attempts. The timeout value between retransmission remains the same as version 1.0.x [31], [40].

The NbTrans value improves redundancy by specifying the number of transmissions for each uplink message. For a confirmed uplink, this specifies how many times retransmission may occur. The default value is one, with a maximum of fifteen. With the default value of one, an ACK will be requested but no retransmission will occur if the ACK is not received. This field forms part of the LinkADRReq command, who is also responsible for the modifications requested by the ADR scheme.

For downlink retransmissions, a new frame counter value must be used and thus the application server must be notified of the failure. This server must decide if a new confirmed downlink frame must be sent to the device. Version 1.1 also specifies that a device will only acknowledge a downlink confirmed frame once. Uplink frames are to be retransmitted NbTrans times and the device must apply frequency hopping between the repeated transmissions. The delay between attempts is left to the end device's developer and any retransmissions should stop once a valid ACK is received.

Retransmissions are not limited to confirmed traffic and can also occur due to certain MAC commands. One such command is the join-request command which the device will retransmit if a response in the form of a join-accept message is not received. The LoRaWAN protocol specifies a retransmission back-off mechanism to prevent network overload during events such as power outages that cause many devices to simultaneously require an acknowledgement or a response to their uplink frames. The mechanism requires that the retransmit interval between frames be random and differ between devices and also adds additional duty-cycle limitations.

IV. IMPACT OF DOWNLINK TRAFFIC ON NETWORK PERFORMANCE

Downlink traffic in a LoRaWAN can be in the form of transmitting an ACK for a confirmed uplink message, sending MAC commands, application layer downlink data or a combination of all of these [41]. Accurate network models are key when determining the impact of downlink traffic on network performance, and modelling should include the PHY and MAC layers. Due to their Aloha like MAC configuration and the DC limits imposed by operating in ISM bands, LoRaWANs suffer from scalability issues [42]. They do, however, perform better than pure Aloha networks, in part due to the capture effect [43], [44] and SF pseudo-orthogonality [13], [45].

Network models evolve and earlier models become outdated as new information on the capture effect and SF orthogonality come to light. Earlier work such as [46], modelled LoRa with the assumption that simultaneous transmissions sent on the same frequency but at different SFs are orthogonal and can thus be simultaneously decoded by the gateway. Research has however shown that the orthogonality is not perfect as should rather be considered quasi-orthogonal. Models should rather consider the interference caused by transmissions using the same SF (co-SF) and different SFs (inter-SF) [13], [34]. Another aspect that models should include is the capture effect. This effect describes the possibility that a packet can successfully be received even if it collides with other packets if its power is sufficiently high [47]. This is problematic for packets from faraway devices, as any collisions with packets sent from nearby devices will result in the far away device's packet not being received.

Even when these factors are considered, authors interpret the protocol differently as can be seen in the case of [44]. In their theoretical model gateways always transmit two acknowledgements (one for each receive window) but this behaviour is not specified in the protocol [48]. This approach to acknowledgements was investigated in [32], which found that transmitting in both windows hurts performance.

In [32], the point is made that all three of the main approaches of evaluating LoRaWAN performance has limitations: experimental work is limited by a large number of variations in cost, size and parameters. Simulation-based approaches are limited by computational power, assumptions made and the level of detail achievable. Mathematical models use ideal assumptions which impact the accuracy, but can be used to focus the efforts of the other options [32].

LoRaWAN simulators have been developed used from a variety of existing WSN platforms such as ns-3, OMNET++ and SimPy. Publicly available LoRaWAN simulators that support downlink traffic are LoRaSim [49], the expanded version of LoRaSim used in [42], the ns-3 modules presented in [50] and [48], LoRaEnergySim [51] and FLoRa [52].

LoRaWAN is very popular under European researchers, and as a result, most simulators are built assuming operation in the EU863-870 MHz band. One exception to this is [53], which simulates 915 MHz North American networks but does not support downlink traffic. This simulator is, however, open-source, and thus can be modified to investigate ACKs. Simulators also need to remain updated with changes to the LoRaWAN protocol as well as relevant research findings such as pseudo-orthogonality and the impact of the Doppler effect on LoRa transmissions as described in [54].

The current LoRaWAN literature focuses almost exclusively only on ACKs during downlink performance evaluations. Notable exceptions to this are the work presented in [55] and [50]. The simulations performed in [55], showed that the addition of downlink data traffic increases the competition experienced by ACK transmissions and as the DC restrictions are tight, this further increases the number of retransmissions. When downlink data traffic was investigated in [50], a small decrease in the Packet Delivery Rate (PDR) for uplink transmissions was noted due to more packets being missed. This is due to the increased time a gateway now spends in transmission mode. Their analysis showed that adding additional gateways can help reduce this negative impact.

The current LoRaWAN literature is also dominated by works examining networks containing only class A nodes. This is logical as class A is expected to be the most common, but does leave the performance of the other classes and networks containing a mix of classes unexplored. One exception to this is the work on class B networks in [56]. They found that the data rate and the number of channels have a significant impact on the delay of downlink traffic.

A. THE IMPACT OF CONFIRMED TRAFFIC

The impact of confirmed traffic arises from different aspects and thus the works examining confirmed traffic were broadly split into several categories. Some works examined more than one aspect and were thus referred to multiple times.

1) THE IMPACT OF NbTrans

To determine why confirmed traffic impacts network performance so severely the authors of [57] build a MATLAB based simulator. They found that as the ratio between confirmed and unconfirmed messages increases, the throughput for both types of traffic decrease.

Increasing the NbTrans value (number of transmissions per uplink) was effective in increasing the packet delivery for confirmed traffic, but came at the expense of unconfirmed traffic's delivery success. Additionally, higher numbers are not necessarily the best as the maximum number of 15 performed the worst as this causes too many collisions. Simulations determined that a value of nine performed best for confirmed traffic. These high numbers may not be usable in public networks as resources must be shared fairly and the cost to unconfirmed traffic might not be acceptable.

When NbTrans was examined in [55], it was found that this number should depend on network size. Simulations using LoRaWANSim found that only a small improvement is seen in small networks. For a 5000 node network, switching from transmitting once to using 5 attempts improved the number of successfully acknowledged frames from 40 % to ≈ 75 %. This increase does however come at a cost, energy consumption rose by nearly 10 % due to the extra energy required when allowing for up to 5 retransmissions attempts.

The evaluation presented in [48], also concluded that a dynamic mechanism is needed and that is should be based on the traffic load. Their ns-3 simulator, which added downlink support to their original work ([58]) was first compared with the analytical model presented in [44]. Unlike the model, the simulator accounts for SF pseudo-orthogonality and the DC restrictions imposed on transmissions.

For small networks (500 devices), increasing the number of transmission attempts was effective in improving the PDR as the DC restrictions are not yet exceeded. Increasing the number of transmission attempts from 2 to 8 allowed a PDR of 95 % to be maintained as the ratio of confirmed traffic is increased from 10 % to 70 %. In large networks (2000 devices), increasing this number no longer allowed a significantly larger percentage of traffic to be confirmed as it also increased the probability of collisions. In large networks, it was found that receiver saturation starts to become the leading cause of packet loss and thus increasing the number of transmissions is no longer beneficial.

The simulator developed in [50] was also used in the LoRaWAN evaluation presented in [59]. Their evaluation focused on the impact of the number of transmissions per packet and the number of available channels. Only two options for the maximum number of transmission attempts are considered (one or eight) and only two for the number of channels available (one or seven). The comparison evaluated 100 % unconfirmed against 100 % confirmed traffic. In a network with seven available channels and under maximum load, unconfirmed traffic's PDR slowly reduces from 100 % to 80 % as the network scales from 1 to 150 devices. In contrast, confirmed traffic's PDR has an aggressive exponential decay and reaches 40 % at 20 devices. This is very aggressive, however, it should be noted that this decay did however not start from 100 % for a single device network, the results of their simulations showed that only a PDR of 80 % was achievable. Increasing the number of retransmissions enables a high PDR of nearly 100 % for unconfirmed traffic, but has no impact on the performance of confirmed traffic.

When a reduced load is considered through a traffic intensity of 10 % the network's performance differs from above. In small one channel networks (less than 40 nodes) confirmed traffic which allowed up to 8 transmissions performed best. Over this 40 device threshold, however, unconfirmed traffic with only 1 transmission remains the best performer. As only one channel is available, increasing the number of retransmissions for unconfirmed traffic drastically reduces the PDR due to the increased collisions it causes. In seven-channel networks, unconfirmed traffic maintained a near 100 % PDR for both maximum transmission options with a less than 5 % difference between them. Confirmed traffic with up to 8 transmissions outperformed only 1 transmission significantly and also maintained this a near 100 % PDR up to 60 devices, after which it declines sharply. The presented results show that there are cases in which confirmed traffic outperforms unconfirmed traffic, but only when the number of channels is limited, the number of devices is low and in networks with moderate loads. Increasing the number of retransmissions can be a method of increasing the PDR for confirmed traffic but does not guarantee performance similar to that of unconfirmed traffic.

As part of investigating the impact of several parameters, the authors of [60] examined NbTrans. They found that increasing this parameter does improve performance for confirmed traffic, but that the increases have diminishing returns. Adjusting this parameter hurt unconfirmed traffic's performance and thus choosing the lowest possible value that meets confirmed traffic's performance requirement is best.

2) CONFIRMED TRAFFIC RATIOS

A mathematical model was presented in [32] and used to study the performance of single gateway networks with 100 % confirmed traffic. Their validation with the ns-3 simulations ([48], [58]) showed that their model is slightly optimistic when predicting the PDR but remains useful. Their further comparison with [44] highlighted that DC restrictions are a key parameter for models to include, as without it the achievable performance will seem much higher.

One of the assumptions made by [32] is that the probability of receiving an ACK sent in the second receive window is 1, the reasoning for this is that the frequency channel chosen for window 2 transmissions are only used for downlink and thus no collisions can occur. This is true for networks in remote areas in which the only traffic on channel 869.525 MHz is the current network. In urban areas, other LoRaWANs will likely be present and this assumption can no longer be made.

A MATLAB simulator, presented in [61], was used to compare scenarios in which nodes require no ACKs, all nodes require an ACK, 25 % require an ACK and 33 % of nodes require an ACK. This simulator only utilised the first receive window when calculating the viability of downlink responses and networks consisted of either 1000 or 2000 end devices. The simulator considered the Packet Error Rate (PER) per node and an average QoS value calculated as the network's total PDR. For the 1000 node network, ACKs did not significantly decrease the average QoS, but requiring an ACK for each uplink increased the percentage of nodes with a PER of between 10 % and 50 % from 1.2 % to 23.9 %. In a 2000 node network, the decrease in QoS and negative impact on packet delivery was much more pronounced. The percentage of nodes with a PER between 10 % and 50 % increase from 6.9 % for no ACKs to 24.5 % when 25 % of traffic require an ACK and when all traffic required an ACK it increased to 70 %.

To investigate bidirectional traffic the authors of [55] created LoRaWANSim, a closed source LoRaWAN simulator based on LoRaSim. Several downlink configurations were tested namely 5 % downlink application layer data (no ACKs), 5 % of uplink traffic require ACKs and finally the combination of the previous two options. As with most evaluations, some simplifications were made. The first was that each device was set to use the maximum data rate allowing it to reach the gateway (similar to the ADR scheme but not dynamic).

Their investigation found that the leading cause of retransmissions was the DC restrictions imposed on gateways [55]. Even when only 5 % of transmissions required an ACK and there was an additional 5 % downlink data, the DC restrictions caused the success of downlink transmissions to drop below 80 % once a network reaches 1000 devices. Any downlink data transmissions aggravate the problem as they cause the gateway to reach its limitations earlier, increasing the number of ACKs it can not transmit.

The network's goodput (application layer throughput) also drops significantly as the percentage of traffic requiring an ACK increases [55]. When 100 % of the traffic required acknowledgement, the goodput drops to 15 % of the level achievable by a network with no acknowledgements. The impact is minimal for low volumes of requests ($\leq$ 5 % require ACKs) but increases to nearly 20 % when 20 % of the traffic requires confirmation. Beyond 20 % of traffic requiring ACKs, the goodput is increasingly negatively impacted.

The impact of traffic ratios was also examined in [48]. In a network where the number of devices is fixed, the PDR was quickly reduced when the percentage of devices using confirmed traffic was increased. In the inverse case where the percentage of devices remains fixed, increasing the number of devices did not significantly decrease the PDR.

To better understand why the PDR is so negatively affected, a scenario in which all devices send confirmed traffic, up to 8 transmissions are allowed and each device sends a packet every 30 minutes was examined closely. It was found that channel impairments had a negligible impact on the PDR and that there is a threshold where interference no longer dominates and the unavailability of receiving paths become the leading cause. It is noted that the addition of more gateways will help to alleviate this problem, but this must be done to optimise the SF allocations within the network. Uniformly spreading gateways will allow more nodes to use the higher data rates, decreasing transmission times and collision probabilities.

A simulator was fed with recorded real traffic traces collected from up to 4 Semtech PicoCell gateways in [41]. When the ratio is below 20 %, the packet loss remains below 10 % and is mainly caused by uplink transmissions being missed due to the half-duplex nature of the gateways. Above this point, DC restrictions become the leading cause of packet loss. In the case of 100 % confirmed traffic, the network loses 86 % of packets, with 76 % of this as a result of their ACKs being unable to be transmitted.

3) THE IMPACT OF SFs

The MATLAB simulator presented in [57] also examined the impact of SFs. Their evaluation of the different SFs found that performance is impacted negatively, due to the additional collisions caused by the increase in transmission time from one SF to the next. Their simulator does have some flaws as it assumes perfect SF orthogonality, does not randomise the time interval between retransmissions (ACK_TIMEOUT) and handles ACK transmissions on the second windows differently. In their setup an ACK is always transmitted with the same SF used by their corresponding uplink regardless of the used window, however, the second window normally uses a fixed value, the suggested is SF12.

The work presented in [48] was followed up with the development of a mathematical model for single gateway networks in [62] and used to study LoRaWAN in [60]. This model was compared with the simulator described in [48] and [58] and was found to match favourably. The impact of using SF12 versus matching the uplink packet's SF was investigated and it was found that by not restricting the gateway more packets could be successfully acknowledged. This effect depends, however, on the traffic volumes. When SF12 is used, the gateway would quickly reach its DC restrictions, keeping it in receive mode. When other SFs are used, the gateway spends more time in transmit mode. Whilst it is thus able to transmit more ACKs, it also missed more uplinks due to the half-duplex nature of the gateway [60].

The mathematical model was presented in [32] also examined the impact of SFs. Their model assumes perfect SF orthogonality and is for a single gateway scenario. Restricting acknowledgements sent using window 2 to the use of SF12 was found to be sub-optimum, as allowing these ACKs to be sent with the same SF as the corresponding uplink improved network performance as this eases meeting the DC restrictions on the gateway [32]. Finally, prioritising reception of packets over the transmission of an ACK resulted in a better PDR. This prioritising was also investigated in [62], who found that this is true of NbTrans is one. Prioritising transmissions will, however, perform better if NbTrans is more than one as this prevents the further retransmission of packets.

A further problem highlighted in [55] and [48], is that the congestion caused by the retransmission mechanism of version 1.0.x networks, detailed in Section III-B. The strategy continually increases the SF used by successive retransmission attempts as it assumes no ACK was received because of distance and does not consider congestion as a cause. Protocol version 1.1 removes this auto-reduction of the used data rate and also allows the number of retransmission to be configurable.

The ns-3 simulator presented in [48] also examined the ADR scheme and found that it tended to assign the same SF to several users and does not consider the resultant loss of SF orthogonality between these nodes. The ACK_TIMEOUT parameter was also evaluated with the simulations performed showing no benefit to this random delay of between one and three seconds. Upon investigation, it was found that collisions occurred amongst nodes using the same SF and that the backoff time is too small for the higher SFs. During retransmissions, the packets would simply collide again as a maximum length SF12 packets take 3.15 s to transmit, which is more than the ACK_TIMEOUT delay.

In [10], several IoT use cases were grouped into one of three groups based on traffic considerations. Out of each group, one use case was selected and network throughput and packet success rate analysed via simulation for confirmed versus unconfirmed SF 7 and SF 12. In terms of network throughput, unconfirmed outperforms confirmed traffic, but the authors note that this result should not be viewed in isolation, instead the PDR should also be examined. The authors examined two scenarios: a small area with a single gateway and a large area covered using multiple gateways. In the first scenario, SF 7 confirmed mode outperformed SF 7 unconfirmed mode up to a point whilst in SF12's case unconfirmed mode remained consistently better than confirmed. As throughput and a result congestion increases, confirmed traffic will start to result in more collisions, causing confirmed mode to no longer outperform unconfirmed mode.

4) RECEIVE WINDOWS

Gateways are required to respond in either of the two receive windows and have to consider the DC restrictions as well as potential interference problems of each window. The first window uses by default the same channel as the original uplink. The use of this window will cause interference with uplink messages from nodes in the network [63].

Most works examining confirmed traffic make use of mathematical models or simulation tools, an empirical approach was however followed in [64]. A gateway and several nodes were placed in an anechoic chamber with the duty cycle restrictions for the nodes selectively disabled. This allowed a single device to emulate 88 to 700 nodes (depending on the SF used). Their experiments found that downlink traffic caused a significant increase in uplink traffic's PER. This is due to the uplink-downlink interference when receive window one is used. Additionally, it was found that the gateway tended to respond using receive window two for high SF traffic and used window one for low SF traffic. This was attributed to the duty cycle limitations that a gateway must adhere to. The time-on-air of high SF ACKs will cause a gateway to quickly exhaust window one, and will thus force the use of window two which has a more relaxed DC requirement.

The second window is on a dedicated downlink only channel, which would be used only by gateways. A network will contain significantly fewer gateways than nodes, making this window a better choice from an interference perspective. However, the DC restrictions applied to each window's channels and the SF used for the response also have to be considered. The use of the second channel tend to occur with higher SFs, the default is SF12, resulting in longer on-air time for transmissions [64]. This will also increase the interference one gateway's transmissions have on another and thus the lowest possible SF should rather be used [32].

The impact of swapping the sub-bands used by each window was investigated in [60].In a mixed traffic network swapping the sub-bands showing a marginal improvement in confirmed packet success ratio, implying that the uplink-downlink interference normally experienced in window one had minimal impact.

A potential solution to eliminate uplink-downlink interference is to disable the use of receive window one and to only rely on window two. This idea is explored in [63], which is discussed in more detail in Section .

5) MULTI-GATEWAY NETWORKS

In a multi-gateway LoRaWAN, the NS is responsible for choosing which gateway must transmit a downlink frame to a node. A gateway selection algorithm is thus required and two approaches are explored in [41]. The first is a simple Signal to Noise Ratio (SNR) based approach currently used by the open-source LoRaWAN NS ChirpStack (previously called LoRaServer) [65]. The recorded SNR values of a node's previous packets are used to select the gateway with which it is best to respond with. If one gateway is preferred over others for a large number of nodes, it could quickly become overworked by the NS. As the algorithm only uses the best gateway and doesn't consider DC restrictions, other gateways, with slightly lower SNR values, end up being underused [41]. The second approach attends to correct this flaw by also considering the DC restrictions placed on gateways.

The addition of more gateways was found to result in a significant improvement, even when the SNR based approach is used. In a quad versus single gateway scenario, the additional gateways halved the packet loss experienced in the network. The main cause of this was that downlink traffic was now split amongst multiple gateways, each with their own DC restrictions, allowing more ACKs to be successfully sent.

The results revealed that this can be improved upon, as with the SNR based approach the first gateway was assigned to transmit 60 % of the ACKs. This gateway was only able to send 24 % of these requests due to DC restrictions, whilst two of the four gateways receives nearly no requests. The second approach, which attempts to balance the load between the gateways, was able to reduce the packet loss by 25 % due to its better load distribution.

An open-source LoRaWAN ns-3 module developed for a scalability analysis is presented in [50]. Part of this analysis was investigating confirmed traffic in a single uplink channel network with a fixed NbTrans of four with either one, two or four gateways. In a one gateway network, confirmed traffic under-performed unconfirmed traffic with the under-performance increasing as the number of devices or the traffic volume increases. The decrease in network PDR is attributed to the number of receive windows missed by the gateway due to DC restrictions. Increasing the number of gateways did improve the PDR, but had a bigger impact on unconfirmed traffic than confirmed traffic. For unconfirmed traffic, adding gateways is very effective as decreases the distance between nodes and their closest gateway, allowing them to use faster data rates and thus decreasing congestion. For confirmed traffic, traffic volumes also play a role as adding gateways does allow more ACKs to be sent but it also allows more packets than previously to be successfully received who now require an ACK. This increases the number of required ACKs, causing DC restrictions to remain a problem. In networks with low traffic volume, additional gateways are very effective, as the volume is low enough that the only bottleneck was DC restrictions.

V. DISCUSSION

The first focus of the discussion is a comparison of the work discussed in the four categories dealing with the impact of confirmed traffic in LoRaWANs. The second focus is a more general analysis of important aspects identified.

A. COMPARISON

Table 2 shows a comparison between the papers discussed earlier. A common theme between evaluations is to only consider the uplink and downlink frequency channels that are considered mandatory by the LoRaWAN protocol. These are the 3 channels used for both downlink and uplink and an additional channel for downlink only. The maximum is 16 channels, [31], and Section 7.2.3 of the ETSI EN300.220 standard ([35]) dictates several sub-bands that may be used by a LoRaWAN. Adding channels in a different sub-band will assist with scalability as DC restrictions are calculated per sub-band. An analysis using e.g. the 10 channels used currently by the TTN or the 12 channels proposed in [63] would provide a result closer to the current performance in deployed large networks.

The table shows several instances in which either 0 % or 100 % confirmed traffic were investigated. Section II highlighted several scenarios in which confirmed traffic was not required by all nodes. Instead, only a small minority of devices in a network will require the use of acknowledgements. Furthermore, acknowledgements are frequently linked

TABLE 2. **Comparison between works examining confirmed traffic.**

Reference	Uplink channels	confirmed to unconfirmed message ratios (%)	DC restrictions	NbTrans
[10]	3	0,100	-	8
[32]	3	100	Yes	1
[41]	-	0-100	Yes	1
[48]	3	0-100	Yes	1, 2, 4, 8
[50]	1	1, 10	Yes	4
[55]	3	5	Yes	1-8
[57]	3	5, 10, 30	Yes	1, 3, 9, 15
[59]	1,7	0, 100	Yes	1, 8
[60]	3	0, 50, 100	Yes	1,4 8
[61]	-	0, 25, 33, 100	No	1
[62]	3	0-100	Yes	1, 2,4, 8
[64]	8	0, 100	Yes	1

to an event occurring, such as cattle breaching a geo-fence. The feasibility of using acknowledgements in a LoRaWAN should thus be seen in the contents of a ratio between confirmed and unconfirmed traffic, with the confirmed traffic potentially occurring in bursts. How well a network scales is thus dependent on this ratio between confirmed and unconfirmed traffic as well as the number of devices, packet size and the transmission frequencies of confirmed and unconfirmed traffic.

Nearly all evaluations did consider the DC restrictions imposed on end-devices and gateways and those that purposefully simulated the impact of NbTrans examined numerous values for this parameter.

B. ANALYSIS

The currently published work examines downlink viability by almost exclusively focusing on the impact of confirmed traffic, i.e. sending ACKs to end devices. However, downlink traffic consists of more than just ACK transmissions, there are MAC command transmissions and application layer data to also consider. Even if confirmed traffic is not allowed by a network operator, downlink traffic would still occur [41]. Improvements in those areas will reduce the total downlink traffic, easing DC restrictions and thus also improving the performance of confirmed traffic.

A potential improvement for downlink traffic is offered in [48], [60], who notes that when end devices open their receive windows, there is the potential that they accidentally lock onto the uplink transmission of another node instead of the gateway's response. This is caused by the preamble of downlink and uplink transmissions being identical, so the receiver is not immediately aware that it is locking onto an uplink transmission.

As the issue of viability only comes into play in large networks, analysis of confirmed traffic has predominately been done using LoRaWAN simulators. The currently available simulators have differences in how the LoRaWAN protocol is implemented. As an example, [57] shows in their Table 1 that ACK_TIMEOUT has a value of 1 s, the LoRaWAN protocol specifies that this should be a random delay between 1 and 3 seconds [31]. In general, simulators are used to investigate a specific topic and thus assumptions/decisions are made on other areas deemed not relevant. Direct comparisons between results are also difficult as different payload lengths, sending intervals, number of nodes, number of channels and so forth are used.

The LoRaWAN protocol could potentially be optimised by modifying the current receive window behaviour. The published work has shown that gateways struggle with the current implementation. Currently, the second window is opened in a downlink only 10 % DC restriction channel. If this channel is rather used for the first receive window, more devices would be able to receive an ACK as not only is due DC restrictions reduced in this channel but there would be no collisions with uplink traffic [48], [64]. Matching the downlink's SF with the original SF of the uplink packet would further improve how many responses can be sent before the 10 % DC restriction is reached.

Whilst the issues raised above must be considered, the existing literature can be summarised to capture what is currently known about the state of confirmed traffic in LoRaWANs and what parameters influence its viability.

Researchers found that sending two ACKs instead of one decreases network performance, receive window 2's use of SF12 is not ideal and that the impact of prioritising reception over transmission at gateways depends on NbTrans and traffic volume [32], [60], [62]. Adjusting the NbTrans parameter can be effective in increasing the PDR but this improvement is limited to small networks [48], [55]. Consensus on the exact threshold after which it becomes negative and what the optimum number of transmissions was not achieved amongst researchers. In large networks, the main cause in the reduction in performance was the DC restrictions imposed on gateways [55]. This is problematic, as nodes are creating a retransmission snowball effect for data that was successfully received. Adding additional gateways can be an effective solution to easing the DC restrictions on gateways, but load balancing between the gateways must be considered [41]. The ACK_TIMEOUT random backoff interval is too short for packets sent using high SFs, they will simply collide again due to their long transmission times [48], [66]. It is good that version 1.1.x of the protocol no longer increase the SF between retransmission attempts, as this only made congestion worst in 1.0.x networks [48], [66].

VI. NEW PROPOSED ACK METHODS

An alternative approach to data collection and acknowledgement is suggested and compared with standard LoRaWAN in [42]. A scheduling scheme called FREE is proposed which requires that nodes store data in bulk and collection takes place periodically. This approach allows devices to send an order of magnitude fewer packets simply by bulking data and sending longer packets. This system is thus not designed for priority data as collection could potentially only occur once every few hours. With FREE, data reception is acknowledged by the gateway using a compressed group ACK scheme.

During their evaluations, nodes generated 20 bytes every 5 minutes, could potentially retransmit a packet up to 8 times and simulations considered 1 day's worth of data. For a standard LoRaWAN using confirmed traffic, the device lifetime drops quickly below two years due to the cost of retransmissions and for a network of 1000 devices, the number of transmissions increased 7.4 times the level of a similar network using unconfirmed instead [42]. LoRaWAN's overall Data Delivery Ratio (DRR) plummets sharply falling below 50 % at ≈ 200 devices as the number of collisions rise sharply. The researchers note that collisions are not the major cause for the drop in DRR, instead, the gateway's DC restrictions prevent it from responding with ACKs. The loss of either the data packet or the ACK due to channel fading also had an impact, but this was much smaller. Their proposed solution outperformed LoRaWAN significantly as the precise scheduling prevented collisions and the DC restrictions were kept in mind.

A method to perform acknowledgement aggregation is proposed in [67]. In this method, the NS periodically sends out an "AggACK" containing ACKs for several devices and received packets. The proposed method was evaluated a modified version of the ns-3 simulator presented in [58]. The results show that this method was able to improve the normalized throughput fand found that transmitting an aggregated ACK every 20 seconds was the optimum waiting time for their evaluated scenario. It is not clear with which SF the aggregated ACK is sent with, or how nodes know on what channel it will be transmitted on. The presented throughput model assumes the number of nodes is large, 1000 nodes, but the ns-3 evaluates only examine up to 200 nodes.

An alternative radio-resource management solution is proposed in [63]. By assigning 869.525 MHz as the designed channel for both downlink windows, collisions between uplink and downlink traffic can be prevented. An operational mode called time-power multiplexing is presented which relies on new hardware designs allowing a gateway to transmit more than one packet at a time. Additionally, new power amplifier subsystems are required to allow for variable transmit power based on the number of simultaneous transmissions [63].

The authors suggest a final modification namely that only one receive window is used, to allow a gateway to maximise the number of downlinks it can group. A comparison, using a MATLAB simulator, between the proposed system and the default protocol showed that higher network capacity is now supported. The improvements did not only improve the capacity for confirmed traffic but also slightly improved unconfirmed traffic's capacity as well. Whilst the scheme did offer improvements, simply reducing the percentage of confirmed traffic would still have a bigger impact on unconfirmed traffic's scalability.

In addition to a transmission scheduling algorithm, the use of group acknowledgements is proposed in [68]. The scheduling algorithm dictates the SF, channel and timeslot for all transmissions thereby allowing a group acknowledgement to be sent for all packets received after each timeslot. The format of this acknowledgement is not described and the proposed solution was compared with ALOHA using simulations. The results showed that the solution can offer an improvement in packet delivery but is dependent on the number of channels. An extension to the LoRaWAN protocol named A2S2-LoRaWAN is proposed in [69]. This extension uses a time-slotted periodic frame structure which is combined with aggregated ACKs to improve the scalability of LoRaWANs. The proposed extension is compared with standard LoRaWAN in simulations, although the gateway is limited to only use one bidirectional channel with a DC restriction of 1 %. This does not quite match the normal operation of an EU based LoRaWAN gateway who would also have access to the downlink only 10 % DC restricted channel used by receive window 2.

A. COMPARISON

Table 3 shows a comparison between the alternative methods to implement ACKs discussed earlier. Only one of the methods was released as open-source, which would allow other researchers to examine the proposed method further. None of the methods was implemented on currently available LoRaWAN hardware, with [63] being unable to do so until a gateway capable of time-power multiplexing is developed.

One of LoRaWAN's strengths is its low power consumption as IoT deployments are frequently in remote and inaccessible areas. The impact of the proposed method on energy consumption is a crucial aspect not all works examined.

VII. OPEN CHALLENGES & POTENTIAL SOLUTIONS

Several challenges must be overcome to improve the viability of confirmed traffic in LoRaWANs. The main challenges found in the literature have been identified and discussed in the section. Aspects of the protocol that are currently statically configured may have to be modified to rather optimally respond to network conditions.

A. INDICATING NETWORK CONGESTION AND REPLY URGENCY

Gateways currently have no method of indicating network congestion to nodes. End devices have the option to delay transmitting an ACK until their next data message, but the gateway does not have a similar option. Nodes are not aware of network congestion so are not instructed to perhaps open a

TABLE 3. Comparison between proposed ACK methods.

Reference	Simulation platform used	Open-source	Examined energy consumption	Implemented
[42] (FREE)	based of Simpy and LoRaSim	Yes	Yes	Future work
[63] (time-power multiplexing)	MATLAB	No	Yes	Requires new hardware
[67] (AggACK)	ns-3 ([58])	No	Future work	No
[68] (GACK)	-	No	No	No
[69] (A2S2-LoRaWAN)	Python	No	Future work	No

third transmission window, after a much longer time interval, or attempt to send less confirmed traffic. This indication of network congestion can also be used by nodes to adjust their transmission frequency or to aggregate their data.

Currently, the LoRaWAN protocol has no method of indicating the urgency of receiving a response for confirmed traffic. The combination of a class A device's capability of only receiving an ACK during two short downlink windows and a gateway's DC restrictions is a big bottleneck as the ACK has to be sent immediately. Should the protocol be adapted to allow end devices to indicate the urgency of an expected acknowledgement this would ease a gateway's DC burden. This indication does not necessarily have to be binary, a value between 0 and 1 can allow gateways to selectively decide which nodes will be answered using their first window versus those who will be acknowledged in their window 2.

B. GROUPING TRAFFIC AND ACK AGGREGATION

Should the confirmation that data is received be important but latency not be grouping several measurements and transmitting them as 1 large payload will have several benefits. Doing so not only has a power consumption benefit [42], [70] but also reduces the number of ACKs sent when compared to sending the data as several smaller packets. The simulations conducted in [42], showed how their scheme's approach of sending fewer but larger packets helped reduce the total number of sent packets significantly (an order of magnitude) which helps reduce the number of collisions.

Acknowledging several devices using one packet can be a method to reduce the number of transmissions required by gateways. This would, however, increase the time-on-air of the transmissions and potentially increase the number of packets a gateway did not receive as it now transmitting for longer periods. This approach would differ from the existing design as the protocol currently aims to minimise the time-on-air of downlink transmissions through steps such as excluding a CRC for the PHYPayload. Should a packet containing a grouped ACK become corrupted it would no longer impact only one node, but rather all of the nodes who were expecting an ACK in this message. How often this would occur, and the impact of the resulting retransmissions, as well as the impact of the increasing time-on-air, is something that needs to be examined further. This is a growing area of LoRaWAN research, with some suggested methods already proposed.

An adaptation of the LoRaWAN protocol to enable broadcasts via a cooperative downlink listening algorithm was proposed in [71]. This method can also be used to broadcasts ACKs to several devices at once thereby reducing the gateway's transmission times. This approach also introduces different methods of reducing traffic volume in a network via groupcasting (requesting data from only certain devices) and geocasting (request that only devices in a certain area transmit). Experiments showed that this method does lead to slightly higher energy consumption in networks where devices don't transmit often, e.g. 1 packet per hour, but the consumption soon matches LoRaWAN when the transmission frequency increases.

C. COMPETITIVE SPACE

The use of Acknowledgements is one way to improve packet delivery in a network, but a big aspect that influences deployments is the fact that an urban area might have multiple LoRaWAN networks competing with one another. Unlike Sigfox, where the network is provided by an operator, anyone can deploy a LoRaWAN network. Creating one network, similar to the Sigfox approach, is however viable with a project such as The Things Network [72] currently boasting approximately 8 000 gateways forming a global network.

Several different networks in one place will frequently receive packets from each other's nodes. If these networks work together, they can reduce the overall load on gateways but this must be done securely. Received packets could be shared with the destined network through services such as the recently announced Packet Broker [73] but this sharing of resources remains a challenge.

D. OTHER CHALLENGES

A challenge when determining the viability of confirmed traffic is how accurate the requirements of a specific use case was captured. More information is required than simply an estimate of how many nodes, a target PDR and expected packet length. Another challenge is that as the protocol is still fairly new, LoRaWAN simulators are not very advanced. They are also not necessarily regularly updated once new research comes to light. Using these tools to examine a complicated project involving a ratio between confirmed and unconfirmed traffic, multiple gateways and the radio environment of a planned deployment site are currently not very feasible.

LoRaWAN gateways have limited scalability as only eight simultaneous signals can be received [47]. This limitation is not always considered by researchers, for example, [57] states their gateways have infinite receive paths. The use of ACKs in large networks cause further scalability issues as LoRaWAN gateways are half-duplex and hence miss uplink messages when transmitting ACKs [41]. In small networks, the traffic volume is low and thus the probability that devices transmit at the exact time interval that an ACK is sent is quite small. This is not the case for large networks, and due to the eight parallel demodulation paths of the gateway, multiple transmission could be lost for each ACK sent. The use of multiple gateways can help alleviate this problem, as the messages can be received by another gateway assuming it is in receive mode. The coverage areas of these gateways should carefully be considered, to ensure adequate overlapping zones to also assist with sharing the load of downlink traffic. The potential performance of full-duplex gateways was investigated in [60], which found them to be very effective in improving uplink traffic's PDR.

LoRaWAN nodes do support multiple channels, although most evaluations stuck to the default channel allocations. The protocol already supports informing nodes of additional channels supported by the gateway. As suggested in [6], some of these additional channels could be dedicated as retransmission only traffic channels for critical packets.

The maximum number of transmissions attempts has been showed to improve the PDR for confirmed and unconfirmed traffic. This increase is seen in small networks and disappears in larger networks. It is however dependent on the traffic volume of a network and not simply on the number of devices. Evaluations examining this configuration parameter typically statically examined this, i.e. fixing the traffic volume and the number of transmission attempts. As suggested in [48], [55], a dynamic system is required in which the maximum number of transmissions attempts are adjusted as the network's traffic volume changes.

There exists an asymmetry in the receiver sensitivity between gateways and end nodes and this could be problematic [48], [60]. A situation can occur in which an end device can receive a gateway using a certain SF, but the gateway would be unable to successfully transmit an ACK using the same SF, as the end device's receiver is not sensitive enough. The gateway is then forced to transmit the ACK using the second receiver window, as this would make use of the maximum spreading factor. This problem can be alleviated by either informing an end device that its first window cannot be used to receive an ACK or inform it to expect a reply using a different SF than it originally used [48].

VIII. CONCLUSION

This paper reviewed several aspects of confirmed traffic in a LoRaWAN. The investigation into IoT use cases revealed that there whilst most IoT traffic can be sent in an unconfirmed fashion, there are many applications for which some degree of confirmed traffic are required.

A detailed analysis of the LoRaWAN protocol revealed that there are aspects of its design such the use of SF12 for transmissions to second receive windows, the random values for ACK_TIMEOUT and the use of NbTrans that can be improved. Overall, the protocol focuses on unconfirmed traffic and lacks some flexibility as well as feedback mechanisms that would improve the viability of confirmed traffic.

Our conclusion is that confirmed traffic is certainly viable in small networks in which data is sent infrequently. In large networks, the DC restrictions which LoRaWAN gateways must adhere to become a severely limiting factor and network performance quickly starts to suffer. This restriction does not necessarily make confirmed traffic in large networks nonviable, it does, however, have to be considered during the design phase of networks. Adding additional gateways is a quick but expensive method to improve the viability, and some improvements can be made to the protocol itself.

Due to the open nature of the protocol, new developments are continually improving on the limitations found in LoRaWANs. At the same time, any long term performance improvements are hard to guarantee for LoRaWANs, as more and more IoT deployments must compete in increasingly crowded frequency bands.

Additionally, the LoRaWAN protocol faces competition as IoT use cases which require acknowledgements may turn to other technologies such as NB-IoT, 802.11ah, 802.11ax and 5G New Radio offerings for Mobile Machine Type Communications (mMTC). These should be compared with LoRaWAN to select the most appropriate solution for each use case.

REFERENCES

[1] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, Nov. 2018.

[2] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on LoRa networking: Research problems, current solutions and open issues," pp. 1–17, 2019, *arXiv:1908.10195*. [Online]. Available: https://arxiv.org/abs/1908.10195

[3] M. A. Ertürk, M. A. Aydin, M. T. Büyükakka lar, and H. Evirgen, "A survey on LoRaWAN architecture, protocol and technologies," *Future Internet*, vol. 11, no. 10, p. 216, Oct. 2019.

[4] M. Saari, A. M. Bin Baharudin, P. Sillberg, S. Hyrynsalmi, and W. Yan, "LoRa—A survey of recent research trends," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2018, pp. 872–877.

[5] M. B. Mollah, S. Zeadally, and M. Azad, "Emerging wireless technologies for Internet of Things applications: Opportunities and challenges," in *Encyclopedia of Wireless Networks*. Basel, Switzerland: Springer, 2019, pp. 1–11.

[6] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017.

[7] M. Asif, S. Khan, R. Ahmad, M. Sohail, and D. Singh, "Quality of service of routing protocols in wireless sensor networks: A review," *IEEE Access*, vol. 5, pp. 1846–1871, 2017.

[8] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, "Evaluating the LoRaWAN protocol using a permanent outdoor testbed," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4726–4733, Jun. 2019.

[9] M. Capuzzo, "Reliable LoRaWAN links: Performance analysis," M.S. thesis, Dept. Inf. Eng., Univ. Padua, Padua, Italy, 2019.

[10] L. Feltrin, C. Buratti, E. Vinciarelli, R. De Bonis, and R. Verdone, "LoRaWAN: Evaluation of link-and system-level performance," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2249–2258, Jun. 2018.

[11] M. O. Farooq and D. Pesch, "Analyzing LoRa: A use case perspective," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Singapore, Feb. 2018, pp. 355–360.

[12] F. Yu, Z. Zhu, and Z. Fan, "Study on the feasibility of LoRaWAN for smart city applications," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Rome, Italy, Oct. 2017, pp. 334–340.

[13] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondón, S. A. Hassan, and M. Gidlund, "Scalability analysis of a LoRa network under imperfect orthogonality," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1425–1436, Mar. 2019.

[14] H. G. S. Filho, J. P. Filho, and V. L. Moreli, "The adequacy of LoRaWAN on smart grids: A comparison with RF mesh technology," in *Proc. IEEE 2nd Int. Smart Cities Conf. (ISC2)*, Trento, Italy, Sep. 2016, pp. 1–6.

[15] F. Orfei, C. B. Mezzetti, and F. Cottone, "Vibrations powered LoRa sensor An electromechanical energy harvester working on a real bridge," in *Proc. IEEE SENSORS*, Orlando, FL, USA, Oct. 2016, pp. 1–3.

[16] A. Nikoukar, S. Raza, A. Poole, M. Güne , and B. Dezfouli, "Low-power wireless for the Internet of Things: Standards and applications," *IEEE Access*, vol. 6, pp. 67893–67926, 2018.

[17] W. Saadeh, S. A. Butt, and M. A. B. Altaf, "A patient-specific single sensor IoT-based wearable fall prediction and detection system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, pp. 995–1003, May 2019.

[18] P. A. Catherwood, D. Steele, M. Little, S. McComb, and J. McLaughlin, "A community-based IoT personalized wireless healthcare solution trial," *IEEE J. Transl. Eng. Health Med.*, vol. 6, 2018, Art. no. 2800313.

[19] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.

[20] A. Mohsin and S. S. Yellampalli, "IoT based cold chain logistics monitoring," in *Proc. IEEE Int. Conf. Power, Control, Signals Instrum. Eng. (ICPCSI)*, Chennai, India, Sep. 2018, pp. 1971–1974.

[21] R. Sanchez-Iborra and M.-D. Cano, "State of the art in LP-WAN solutions for industrial IoT services," *Sensors*, vol. 16, no. 5, p. 708, May 2016.

[22] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, "Low throughput networks for the IoT: Lessons learned from industrial implementations," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Milan, Italy, Dec. 2015, pp. 181–186.

[23] 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.

[24] L. Leonardi, F. Battaglia, G. Patti, and L. L. Bello, "Industrial LoRa: A novel medium access strategy for LoRa in Industry 4.0 applications," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc.*, Washington, DC, USA, Oct. 2018, pp. 4141–4146.

[25] R. M. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, and T. M. Chen, "Optimal policy derivation for transmission duty-cycle constrained LPWAN," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3114–3125, Aug. 2018.

[26] mOOvement. *GPS Ear Tags for Cattle Tracking*. Accessed: Dec. 7, 2019. [Online]. Available: https://www.moovement.com.au/

[27] J. Navarro-Ortiz, J. J. Ramos-Munoz, J. M. Lopez-Soler, C. Cervello-Pastor, and M. Catalan, "A LoRaWAN testbed design for supporting critical situations: Prototype and evaluation," *Wireless Commun. Mobile Comput.*, vol. 2019, Feb. 2019, Art. no. 1684906.

[28] N. Maalel, E. Natalizio, A. Bouabdallah, P. Roux, and M. Kellil, "Reliability for emergency applications in Internet of Things," in *Proc. IEEE Int. Conf. Distrib. Computing Sensor Syst. (DCoSS)*, Cambridge, MA, USA, May 2013, pp. 361–366.

[29] A. Dvornikov, P. Abramov, S. Efremov, and L. Voskov, "Qos metrics measurement in long range iot networks," in *Proc. IEEE 19th Conf. Bus. Inform. (CBI)*, vol. 2, Thessaloniki, Greece, Jul. 2017, pp. 15–20.

[30] D. H. Kim, J. B. Park, J. H. Shin, and J. D. Kim, "Design and implementation of object tracking system based on LoRa," in *Proc. Int. Conf. Inf.Netw. (ICOIN)*, Da Nang, Vietnam, Jan. 2017, pp. 463–467.

[31] *LoRaWAN 1.1 Regional Parameters*, LoRa Alliance, Fremont, CA, USA, Jan. 2018, pp. 1–72.

[32] M. Capuzzo, D. Magrin, and A. Zanella, "Mathematical modeling of LoRaWAN performance with bi-directional traffic," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 206–212.

[33] *LoRaWAN Specification V1.1*, LoRa Alliance, Fremont, CA, USA, Oct. 2017, pp. 1–101.

[34] R. B. Sørensen, N. Razmi, J. J. Nielsen, and P. Popovski, "Analysis of LoRaWAN uplink with multiple demodulating paths and capture effect," in *Proc. IEEE Int. Conf. Commun.*, Shanghai, China, May 2019, pp. 1–6.

[35] *European Telecommunications Standards Institute*, document ETSI EN 300 220-1 V2.4.1, Feb. 2012, pp. 1–73.

[36] *ERC Recommendation 70-03*, Standard 70–03, CEPT, Jun. 2019, pp. 1–78.

[37] The Things Industries. *LoRaWAN Frequencies Overview*. Accessed: Jul. 19, 2019. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/frequency-plans.html

[38] *Semtech SX1276/77/78/79 Datasheet*, Semtech, Camarillo, CA, USA, Mar. 2015, rev. 4.

[39] *LoRaWAN Specification V1.0.3*, LoRa Alliance, Fremont, CA, USA, Jul. 2018, pp. 1–72.

[40] *LoRaWAN 1.0.3 Regional Parameters*, LoRa Alliance, Fremont, CA, USA, Jul. 2018, pp. 1–71.

[41] V. Di Vincenzo, M. Heusse, and B. Tourancheau, "Improving downlink scalability in LoRaWAN," in *Proc. IEEE Int. Conf. Commun.*, Shanghai, China, China, May 2019, pp. 1–7.

[42] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, B. O'Flynn, and D. Pesch, "FREE–fine-grained scheduling for reliable and energy efficient data collection in LoRaWAN," pp. 1–15, 2018, *arXiv:1812.05744*. [Online]. Available: https://arxiv.org/abs/1812.05744

[43] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proc. ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Malta, Malta, Nov. 2016, pp. 59–67.

[44] D. Bankov, E. Khorov, and A. Lyakhov, "Mathematical model of LoRaWAN channel access with capture effect," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017, pp. 1–5.

[45] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, p. 1193, May 2017.

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

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

[48] M. Capuzzo, D. Magrin, and A. Zanella, "Confirmed traffic in LoRaWAN: Pitfalls and countermeasures," in *Proc. 17th Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Capri, Italy, Jun. 2018, pp. 1–7.

[49] T. Voigt, M. C. Bor, U. Roedig, and J. M. Alonso, "Mitigating internetwork interference in LoRa networks," in *Proc. Int. Conf. Embedded Wireless Syst. Netw. (EWSN)*, Uppsala, Sweden, Feb. 2017, pp. 323–328.

[50] F. Van Den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017.

[51] G. Callebaut, G. Ottoy, and L. Van der Perre, "Cross-layer framework and optimization for efficient use of the energy budget of IoT Nodes," pp. 1–7, 2018, *arXiv:1806.08624*. [Online]. Available: https://arxiv.org/abs/1806.08624

[52] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of LoRa networks for dense IoT deployments," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Taipei, Taiwan, Apr. 2018, pp. 1–9.

[53] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi, "Throughput, coverage and scalability of LoRa LPWAN for Internet of Things," in *Proc. IEEE/ACM 26th Int. Symp. Qual. Service (IWQoS)*, Banff, AB, Canada, Jun. 2018, pp. 1–10.

[54] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 3, Mar. 2017, Art. no. 155014771769941.

[55] A.-I. 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.*, Singapore, Dec. 2017, pp. 1–6.

[56] F. Delobel, N. El Rachkidy, and A. Guitton, "Analysis of the delay of confirmed downlink frames in class B of LoRaWAN," in *Proc. IEEE Veh. Technol. Conf.*, Sydney, NSW, Australia, Jun. 2017, pp. 1–6.

[57] M. Centenaro, L. Vangelista, and R. Kohno, "On the impact of downlink feedback on LoRa performance," in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2018, pp. 1–6.

[58] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, Apr. 2017, pp. 1–7.

[59] F. H. Khan and M. Portmann, "Experimental evaluation of LoRaWAN in NS-3," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Sydney, NSW, Australia, Nov. 2019, pp. 1–8.

[60] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of LoRaWAN performance under different parameter settings," *IEEE Internet Things J.*, to be published.

[61] N. Varsier and J. Schwoerer, "Capacity limits of LoRaWAN technology for smart metering applications," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, May 2017.

[62] D. Magrin, M. Capuzzo, A. Zanella, and M. Zorzi, "A complete LoRaWAN model for single-gateway scenarios," 2019, *arXiv:1912.01285*. [Online]. Available: https://arxiv.org/abs/1912.01285

[63] M. Centenaro and L. Vangelista, "Time-power multiplexing for LoRa-based IoT networks: An effective way to boost LoRaWAN network capacity," *Int. J. Wireless Inf. Netw.*, vol. 26, no. 4, pp. 308–318, Dec. 2019.

[64] K. Mikhaylov, J. Petajajarvi, and A. Pouttu, "Effect of downlink traffic on performance of LoRaWAN LPWA networks: Empirical study," in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Bologna, Italy, Sep. 2018, pp. 1–6.

[65] O. Brocaar. *Open-Source LoRaWAN Network Server Stack*. Accessed: Dec. 11, 2019. [Online]. Available: https://www.chirpstack.io/

[66] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of LoRaWAN channel access," in *Proc. Int. Conf. Eng. Telecommun.*, Moscow, Russia, Nov. 2016, pp. 10–14.

[67] Y. Hasegawa and K. Suzuki, "A multi-user ACK-aggregation method for large-scale reliable LoRaWAN service," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.

[68] J. Lee, W. C. Jeong, and B. C. Choi, "A scheduling algorithm for improving scalability of LoRaWAN," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 1383–1388.

[69] G. Yapar, T. Tugcu, and O. Ermis, "Time-slotted ALOHA-based LoRaWAN scheduling with aggregated acknowledgement approach," in *Proc. FRUCT*, Helsinki, Finland, Nov. 2019, pp. 1–8.

[70] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, p. 2364, Oct. 2017.

[71] B. Kim and K.-I. Hwang, "Cooperative downlink listening for low-power long-range wide-area network," *Sustainability*, vol. 9, no. 4, p. 627, Apr. 2017.

[72] The Things Industries,. *The Things Network*. Accessed: Jul. 10, 2019. [Online]. Available: https://www.thethingsnetwork.org/

[73] *Packet Broker*. Accessed: Jul. 10, 2019. [Online]. Available: https://www.packetbroker.org/

JACO MORNÉ MARAIS (Student Member, IEEE) received the B.Eng. and M.Eng. degrees from the University of Pretoria, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree. His research interests include LoRaWAN, low-power wide area networks, wireless sensor networks, and embedded systems.

ADNAN M. ABU-MAHFOUZ (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees in computer engineering from the University of Pretoria, Pretoria, South Africa, in 2005 and 2011, respectively. He is currently a Principal Researcher with the Council for Scientific and Industrial Research (CSIR), a Research and Innovation Associate with the Tshwane University of Technology, Pretoria, South Africa, and an Extraordinary Faculty Member with the University of Pretoria. He participated in the formulation of many large and multidisciplinary Research and Development successful proposals (as a Principal Investigator or main author/contributor). He is also the Founder of the Smart Networks Collaboration Initiative that aims to develop efficient and secure networks for the future smart systems, such as smart cities, smart grid, and smart water grid. His research interests include wireless sensor and actuator networks, low-power wide area networks, software defined wireless sensor networks, cognitive radio, network security, network management, sensor/actuator node development, smart grid, and smart water systems. He is a member of many IEEE technical communities. He is an Associate Editor of IEEE Access, the IEEE Internet of Things, and the IEEE Transactions on Industrial Informatics.

GERHARD P. HANCKE (Life Fellow, IEEE) received the B.Sc., B.Eng., and M.Eng. degrees in electronic engineering from the University of Stellenbosch, South Africa, in 1970 and 1973, respectively, and the D.Eng. degree from the University of Pretoria, South Africa, in 1983. He is currently a Professor with the University of Pretoria. He is also recognized internationally as a pioneer and the leading scholar in industrial wireless sensor networks research. He co-edited a textbook, *Industrial Wireless Sensor Networks: Applications, Protocols and Standards* (2013), the first on the topic. He has been serving as an Associate Editor and a Guest Editor for the IEEE Transactions on Industrial Informatics, IEEE Access, and the IEEE Transactions on Industrial Electronics. He is also the Co-Editor-In-Chief of the IEEE Transactions on Industrial Informatics and a Senior Editor of IEEE Access. He initiated and co-edited the first special section on industrial wireless sensor networks in the IEEE Transactions on Industrial Electronics, in 2009, and the IEEE Transactions on Industrial Informatics, in 2013.