Lightweight Timeslot Scheduling Through Periodicity Detection for Increased Scalability of LoRaWAN

Joseph Finnegan
Department of Computer Science
Maynooth University
Maynooth, Ireland
Email: joseph.finnegan@mu.ie

Ronan Farrell
Department of Electronic Engineering
Maynooth University
Maynooth, Ireland
Email: ronan.farrell@mu.ie

Stephen Brown
Department of Computer Science
Maynooth University
Maynooth, Ireland
Email: stephen.brown@mu.ie

***Abstract*—Massive Machine Type Communications is a wireless paradigm which focuses on traffic that is transmitted by a huge number of low cost, low power, infrequently transmitting devices. LoRaWAN, a Low Power Wide Area Network technology, is particularly suited to contribute to coverage of this form of traffic. In this work, we introduce a novel, lightweight timeslot scheduling scheme that supports the requirements for massive Machine Type Communications on LoRaWAN networks (based on traffic periodicity, and the multiple channels and quasi-orthogonal data rates in LoRaWAN). Our novel approach does not require extended downlink transmissions from the gateways, and does not require time synchronisation between the devices and the LoRaWAN Network Server. We implement our scheme in a publicly available LoRaWAN ns-3 module. Our results show that the scheme doubles the number of frequently transmitting mMTC devices that can be handled by a single LoRaWAN gateway while providing the same level of performance in terms of successful packet deliveries, and maintaining a reasonable delay for mMTC use cases, without impacting the ability of the network to send downlink frames or acknowledge high priority packets. Our contribution is a novel approach to TDMA that is suited for mMTC networks.**

***Index Terms*—LoRaWAN, LPWAN, IoT, Traffic Categorisation, TDMA**

I. Introduction

The developing support of massive machine-type communications (mMTC) is one of the defining characteristics of 5G. mMTC focuses on traffic that is transmitted by a huge number of devices, where the individual amount of traffic from each device is low. Low Power Wide Area Network (LPWAN) technologies, such as LoRaWAN, are particularly suited to contribute to coverage of mMTC use cases [1].

mMTC differs significantly from Human-Type Communications (HTC). mMTC features a large amount of heterogeneous devices simultaneously connected to the same gateway (with a targeted connection density of 1 million devices per km^2). mMTC devices are typically low-cost, energy constrained, and ultra low power [2]. mMTC traffic is uplink-dominant, packets are generally shorter, devices are generally less mobile, and there are longer periods in-between transmissions [3], [4]. mMTC traffic is also generally distributed throughout the day, instead of having set peak hours [5]. Whereas HTC traffic is heterogeneous, mMTC traffic is highly homogeneous (all machines running the same application behave similarly). In addition, whereas HTC traffic is uncoordinated on small timescales, mMTC may be coordinated (many machines react on global events in a synchronised fashion).

Analysis of mMTC traffic has led to the development of mMTC traffic models. 3GPP have developed a MTC traffic model consisting of the two scenarios [6], which categorises traffic into two types: 1) periodic: deterministic traffic that is transmitted at regular intervals, and 2) event-driven: asynchronous traffic that is triggered by some external event. Note that a single device may transmit both periodic and event-driven traffic (for example, a rainfall monitor that reports an alert when the reading exceeds a threshold value) [3]. In addition, event-driven traffic may be independent to a single device, or may be geographically correlated (for example, flood alerts) [7]. Spacial and temporal correlated events can result in bursty traffic which impairs network performance [8].

The transmission time of individual LoRaWAN devices (including LoRaWAN gateways) is limited by the duty cycle regulations of the sub-1GHz ISM bands. As a result, LoRaWAN is not suitable for applications with strong real-time or low latency requirements, and downlink traffic and confirmed uplink frames are limited. In addition, this means that control plane traffic, used in the management of a LoRaWAN network, is limited. LoRaWAN manages this limitation through the use of an ALOHA-based channel access method, which does not organise any slots in frequency or time for devices in the network, and thus requires minimal downlink control plane traffic for connected devices.

However, as previously described, mMTC traffic transmitted over LoRaWAN can generally be described as periodic or event-driven. In this work, we leverage the periodicity of LoRaWAN traffic to increase the scalability of the LoRaWAN network. As LoRaWAN is already unsuitable for applications with strict low latency requirements, an increased delay to

978-1-7281-7374-0/20/$31.00 ©2020 IEEE

DOI 10.1109/WoWMoM49955.2020.00017

a certain degree is not a major concern. In addition, as LoRaWAN features a minimum of 3 separate channels, and 6 quasi-orthogonal data rates, co-located devices that are programmed to continuously transmit simultaneously will not always interfere with each other. These factors enable the detection of conflicting periodic traffic in LoRaWAN networks.

The scheme in this work uses the LoRaWAN Network Server to learn the characteristics of incoming traffic from individual devices, uses these to make predictions about future network traffic arriving at each gateway in the network, and ultimately reduce expected collisions from periodic traffic of geographically co-located devices. We develop a system that is suited to the characteristics of mMTC and LoRaWAN (in particular, periodicity, and LoRaWAN multiple channels and quasi-orthogonal data rates), which enables lightweight timesslot scheduling that increases scalability without introducing strain on downlink feedback from LoRaWAN gateways. As far as we are aware, this mMTC traffic model-based system is a novel approach for TDMA over LoRAWAN.

We implement our proposal, which we call LTS (Lightweight Timeslot Scheduling) in a previously openly released LoRaWAN ns-3 module [9]. Our results show that the scheme doubles the number of frequently transmitting mMTC devices (once every 10 minutes) that can be handled by a single LoRaWAN gateway while providing the same level of PDR. The scheme provides these gains while maintaining a tolerable delay for mMTC use cases, and without impacting the ability of the network to send downlink frames or acknowledge high priority packets.

This paper is structured as follows: Section II provides an overview of related work. Section III describes the proposed method. Section IV provides simulated results. Section V concludes the paper.

II. Related Work

Note that the designation mMTC was first formally defined by 3GPP and ITU in 2018 as a subset of MTC (i.e. massive MTC, as opposed to critical MTC which has stricter latency requirements) [1]. The majority of research described below pre-dates the new designation and simply refers to MTC. However, the researchers generally do model MTC traffic that is much more similar to mMTC rather than critical MTC (no major focus on low latency) and as such can be considered related to mMTC.

A. mMTC Traffic Modeling

More advanced traffic modeling of MTC than the 3GPP model has been performed by [10], which models individual devices using Markov Modulated Poisson Processes, and geographically co-located event-based through interactions between the processes. [8] evaluate the performance of LoRaWAN using the previously mentioned model which combines periodic and event-based traffic, and show that LoRaWAN is unable to handle significant temporally correlated event-based traffic. [11] analyses LoRaWAN performance under different traffic models based on different IoT use cases, and demonstrates the limitation of scalability for the slowest LoRaWAN data rate.

[12] also extend the 3GPP model by integrating an open telecommunications dataset into the MTC traffic model. A comparison of Machine-to-Machine (M2M) traffic models against real world data sets is performed by [7], who do find that the majority of M2M traffic is deterministic and periodic. An analysis of M2M traffic on cellular traffic is performed by [13]. [14] study the loss rate and delay of non-Poisson M2M traffic in LTE networks. In [3], the characteristics of aggregated periodic IoT data from related work is analysed and compared to a Poisson process as an approximation for the traffic. Finally, [15] propose a model of MTC traffic at the central controlling node of a Wireless Sensor Network, and study packet loss and delay for varying arrival rates.

B. mMTC-Traffic Modeling Aware MAC

Consideration of the traffic patterns of mMTC have previously been used in order to enable greater network performance. In [16], categorisation of traffic of a generic LPWAN device is performed to enable reduced latency for urgent and event-driven traffic, by providing one dedicated contention-based channel for event-based and urgent packets, and another dedicated contention-free channel for periodic traffic.

In [17], the researchers focus on enabling a reduced latency for event-based traffic in IEEE 802.11ah. This is achieved through the use of a pool of reserved slots which changes in size based on recent reporting demands. Thus, mMTC traffic patterns are considered to enable greater performance, but gains are made through consideration of event-based traffic, not periodic. [18] provide a scheduling method for fast uplink grant transmissions in a cellular system for MTC based on multi-arm bandits.

Other research has focused on the graceful support of heterogeneous traffic policies in a single network. [19] introduce and analyse scheduling policies for heterogeneous traffic over LTE, including MTC traffic. In [20], time allocation for a network of heterogeneous devices is formulated as a non-cooperative game, factoring in heterogeneous requirements and capabilities of devices in the network. [21] proposes a system that monitors traffic patterns from networks consisting of devices with multiple wireless access options, and modifies the selected wireless technology of a device to the most suitable option based on the analysis.

C. LoRaWAN Enhancements for Scalability

Other approaches for increasing the scalability of LoRaWAN networks are available in the literature. In [22] beacon frames are introduced to indicate the allowed data rates and RSS limit of each channel for a following beacon period. This effectively groups devices by distance from the gateway, minimising the capture effect. In [23], TDMA and a latency-tolerant packet-aggregating access scheme are implemented in LoRaWAN in order to increase efficient channel utilisation. [24] combines wake-up radios and a cluster-based topology with LoRaWAN in order to introduce on-demand TDMA, enabling the Network Server to directly request data from individual devices and thus organise scheduling to prevent collisions.

Other approaches towards increasing scalability maintain the ALOHA-based channel access of LoRaWAN while optimising the data rate allocation of devices according to some metric. [25] allocate data rates in order to achieve an equal collision probability across all devices, [26] aim to the balance of the link load across channels, and [27] aim to optimise the overall throughput. [28] introduce the use of mesh topologies in longer range networks to enable the avoidance of use of slower, less energy efficient data rates.

Overall, though TDMA-like approaches have been applied to LoRaWAN before, our work is novel because it takes into account the characteristics of mMTC traffic to provide scheduling that does not require extended downlink transmissions from the gateways, and does not require time synchronisation between the devices and the LoRaWAN Network Server. LTS can also be easily integrated into the LoRaWAN specification through the addition of just one new set of MAC commands.

III. Approach

This section describes the step-by-step methodology of our proposed approach, LTS. The method consists of five steps:

A. Data Collection
B. Periodicity Detection
C. Collision Prediction
D. Collision Avoidance
E. Decision Propagation

LTS is run on the Network Server of the network, for each individual LoRaWAN Gateway. The method is ran once every m seconds, which is a configurable parameter. In this work we assume the method is ran once every hour, and focus on frequently transmitting LoRaWAN devices (i.e. every device transmitting several times per hour); the method could equally be applied in a network with seldomly transmitting devices (e.g. once every six hours, or once per day) by instead running the method less frequently and thus over a longer timeframe (e.g. once per day, or once per week). All of the processing for the method takes place on the Network Server, and thus the method is not dependent on the processing abilities of individual LoRaWAN devices or gateways.

A. Data Collection

The initial step is to record the necessary information required to identify periodicity. In particular, for each device in the network, the Network Server maintains a record of the time received, Gateway, data rate, packet length, and channel of each received uplink frame.

B. Periodicity Detection

The next step is to analyse the recorded data to classify each recorded frame as periodic or event-based. For the time period being analysed, the activity of each LoRaWAN device is represented by a separate discrete bit sequence $S = \{t_0, t_1, ...t_{n-1}\}$, with size n, where the ith bit of the sequence represents the activity of the device in the time between $i * q$ and $(i+1) * q$, where q is the time taken to transmit a packet of a set length using the current data rate of the device. This timeslot length for each data rate is a configurable parameter, but should be long enough to prevent the transmission time of a LoRaWAN packet exceeding the length of two timeslots. As previously mentioned the frequency that the method is run is also a configurable parameter; thus the value of n is calculated based on this value m and the timeslot length q:

$$n = m/q \tag{1}$$

Then, for each timestamp, $t_i = 1$ if a packet has been received by the Network Server from this device in timeslot i, and otherwise $t_i = 0$. The goal is thus to identify the periodic patterns in this sequence, where event-based traffic effectively represent false positive readings, and missed uplink frames (due to collisions or otherwise) represent false negative readings.

Autocorrelation of the timeslot sequence is used to find candidate solutions C across the entire search space: autocorrelation of the entire sequence reveals candidate periodicity values, and then the correlation of a sequence with that found periodicity (and an offset of 0) with the original sequence reveals candidate offset values. This is a well known method for periodicity detection in binary sequences, but struggles to accurately detect exact period and offset values [29]. Thus, the approach presented by [29] is then used to score the subset of solutions in the vicinity of found candidate solutions. This approach is suitable for sequences with multiple periodic patterns and low sampling rates, and is less sensitive to noise than approaches based on the Fourier transform or autocorrelation. The system is based on a score function:

$$score = (1-\alpha)(\frac{|S_T C_T|}{S_T}) - \alpha(\frac{|S_F C_T|}{S_F}) \tag{2}$$

where S_T is the number of 1s in S, S_F is the number of 0s in S, $|S_T C_T|$ is the number of elements of the sequences S and C that are both 1, and $|S_F C_T|$ is the number of elements of the sequences S and C where the value in C is 1, but the value in S is 0.

This function thus scores potential periodic patterns by their closeness to the dataset in terms of false positives and negatives. Timeslot members of candidate solutions that score above a threshold α are filtered out of the next iteration of the search, and the search continues until the highest scoring candidate solution has a score less than the threshold. With a properly chosen α, periodic patterns with some missing transmissions will still be detected, and event-driven traffic will result in poorly scoring periodic patterns and thus will not be detected as periodic data.

This detection method is computed on the data collected for each device in the network, and provides an output of a set of tuples of candidate solutions in the form (p, o, id), where p is the periodicity, o is the offset of the sequence, and id is the ID of the device.

C. Collision Prediction

Step B) results in set of tuples for each device indicating the expected periodic traffic. Each sequence of a device using the same data rate will be the same length. The next step is to identify devices that have periodic transmissions that overlap in time and data rate. i.e. for two devices that overlap in data rate and location, calculate the overlap in their generated sequence. If the overlap exceeds some threshold, then the offset of one of the device's traffic will be modified to prevent later collisions.

Namely, across the periodicity set $\{(p_0, o_0), \ldots (p_N, o_N)\}$, integer solutions are found for multiples of p_i and p_j that are equal to one another when considering the relative offset i.e. when $ap_i + bp_j = o$, where a and b are integers and $o = |o_i - o_j|$. This is the simplest form of the Diophantine equation, and is known to have a solution[30].

Firstly, we define $d = gcd(p_i, p_j)$. If o is a multiple of d, then if the sequences (p_i, o_i) and (p_j, o_j) are generated there will be an eventual shared member of the sets. If o is not a multiple of d i.e. $o\%d \neq 0$, then the two sequences will never overlap.

Example 1:

$(50, 1), (20, 2)$

$d = gcd(50, 20) = 10$

$o = |2 - 1| = 1$

Therefore o is not a multiple of d. If we generate the sequences:

$(50, 1) = [1, 51, 101, 151, 201...]$

$(20, 2) = [2, 22, 42, 62, 82...]$

It can be seen that indeed the two sequences will never overlap.

Example 2:

$(50, 1), (20, 11)$

$d = gcd(50, 20) = 10$

$o = |1 - 11| = 10$

Therefore o is a multiple of d. If we generate the sequences:

$(50, 1) = [1, 51, 101, 151, 201...]$

$(20, 11) = [11, 31, 51, 71, 91...]$

It can be seen that the two sequences do overlap.

Then, if two overlapping sequences are found, the percentage of each sequence's members that will overlap with the other sequence can be calculated using $\frac{p1}{lcm(p1,p2)} * 100$ and $\frac{p2}{lcm(p1,p2)} * 100$ respectively.

From Example 2 it can be seen that the first overlap of the sequence occured at 51. Then if $lcm(50, 20) = 100$ is calculated, this shows that the sequence of overlapping members of the sets is given by the equation $51 + 100k$, and that the percentage of members of the $(50, 1)$ sequence that overlap is $\frac{50}{lcm(50,20)} * 100 = 50\%$.

D. Collision Avoidance

Now, the principles outlined in step B) are used to change the offset of periodicities to prevent future collisions between devices. Note that in (p_i, o_i) periodicities, the p values cannot be changed. Therefore, the gcd and lcm values between pairs of sequences also cannot be changed i.e. two sequences either 1) do collide, and collide by a certain percentage of elements, or 2) never collide. Thus the goal is to modify the o values between pairs of sequences to minimise the number of o values that are a multiple of the pair's d value.

The collision avoidance algorithm is described in Algorithm 1. The heuristic algorithm works by building a vector for each data rate describing the usage of each timeslot, and then modifying o_i values to balance the usage of timeslots.

Such a timeslot scheduling approach introduces delay for all future transmissions from a device. A general accepted latency for mMTC is 10 seconds [1], and thus this is maintained as the maximum delay which the algorithm can impose. The max_change variable reflects the maximum number of timeslots that a device can be shifted while still maintaining below this level of delay. This variable thus is relative to the size of the timeslot for the data rate; an example of max_change values is provided in the next section.

In addition, we add the constraint that a device can only be assigned a timeslot which is after it's initial o_i value, with the logic that the reason for a transmission may be time-specific, and thus this approach enables devices to wake to take a sensor reading, then return to sleep and wake once again to transmit at the assigned time.

Note that the changes to the o_i values are maintained across time, and that the max_change value is relative to the initial o_i value that the device had at the very first time the algorithm was run for this device.

The output of the algorithm described in this section is a change of o_i value, designated c, for a subset of (p_i, o_i) that will reduce future collisions across devices.

Algorithm 1 Collision Avoidance Algorithm

```
once every period:
for each LoRaWAN data rate do
    sort (p,o) tuples by p, then o
    create a vector of size timeslots(DR), of all 0s
    for each (p,o) do
        increment used slots in the vector
    for each (p,o) do
        count number of overlapping slots with this (p,o)
        (not including own transmissions)
        divide by number of transmissions
    if overlapCount ≥ 1 then
        repeat count for new o values of {o ≤ x ≤ o + max_change}
        (i.e. the potential slots that could be used)
    if lowestOverlap < overlapCount then
        modify the o value of this (p,o) to fit new slot
        modify the initial vector to reflect this change
```

E. Decision Propagation

Finally, we define a LoRaWAN MAC command and response which can be piggybacked onto the next downlink frame for a device, which instructs the device to delay all

Size (bytes)	1
TimeslotDelayReq Payload	timeslotDelay

Fig. 1. Format of the TimeslotDelayReq MAC Command

Size (bytes)	1
TimeslotDelayAns Payload	Status

Fig. 2. Format of the TimeslotDelayAns MAC Command

future uplinks by c timeslots, which corresponds to a set amount of time dependent on the LoRaWAN data rate of the device. The original channel selection method of LoRaWAN is maintained i.e. channel selection is performed on a random basis from any channel that is available for immediate transmission without breaking the duty cycle regulations of the $EU868$ band.

The format of these new MAC commands are provided in Figures 1 and 2. The TimeslotDelayReq MAC command, sent only in downlink frames, contains a single byte which represents the number of timeslots a device is to delay all future transmissions by. A check is performed both on the Network Server and the device to prevent the device from choosing a timeslotDelay value that will result in an introduced delay that is greater than the accepted maximum tolerable delay for mMTC communications of 10s. The TimeslotDelayAns MAC response, sent only in uplink frames, contains a single byte, the low order bit of which indicates whether the device could successfully change the timeslot delay. The remaining seven bits are reserved for future use.

Note that each new downlink frame generated does put additional strain on the gateway (and can impact the packet delivery rate as current LoRaWAN gateways operate in half-duplex). However, the LoRaWAN Adaptive Data Rate scheme for maintaining optimal data rate (use of which is highly recommended by the LoRaWAN specification) does already require the transmission of 1 downlink frame every 32 uplink frames. The generated MAC commands from the described algorithm can be piggybacked on those frames.

Runs of LTS are computed only on data gathered since the last iteration. Clock drift of devices will thus be handled automatically based on the received time of recent frames, with new $timeslotDelay$ values allocated as required. Thus, there is not an accumulative effect from clock drift, and time synchronisation of devices is not required.

IV. Simulation

We implement LTS in ns-3, and compare it to a previously released implementation of LoRaWAN [9]. The simulations consist of a single LoRaWAN network, where N LoRaWAN Class A devices are equally distributed across a disk of radius 4km. Devices are assigned an appropriate LoRaWAN data rate based on the distance from the gateway, representing an overall topology which would be reached as a result of each device using the LoRaWAN Adaptive Data Rate scheme. The key parameters in the simulations are outlined in Table I. Each simulation is ran five times, using a different random stream. Devices transmit initially at a uniformly random time between 0 and $Uplink\_Period$, and then transmit once every $Uplink\_Period$. This is equivalent to the periodic, deterministic traffic modelled in [6]. Note that the periodicity of a transmission every 600s is actually very frequent for mMTC use cases [1], and thus in terms of number of devices supported the results shown can be considered a difficult case scenario.

TABLE I
Simulation Parameters

Gateways	1
End Devices	100, 200, 400, ... 5600
Disk Radius	4000m
Uplink Period	600s
Downlink Period	No data plane DL
Algorithm Frequency	once per hour
Packet Size	20 bytes (excluding header)
Random Streams	5
Simulation Time	150 * Uplink Period

For simplicity, it is assumed that all devices are sending frames of the same length, and that the network features only one Gateway node. Every uplink packet sent is 33 bytes long, corresponding to 13 bytes for the LoRaWAN header and 20 bytes of application-layer payload.

The length of a timeslot for $DR0$ is calculated to be the length of time to send a frame of this length. The length of a timeslot for each other data rate is calculated to be half the length of the transmission time of the next slowest data rate, which is an approximation of the difference in time taken to transmit two equivalent LoRaWAN packets using two different data rates. As previously mentioned, we assume the algorithm runs once per hour. Thus, the length of a timeslot, and thus the number of timeslots and max_change value for each data rate is provided in Table II.

mMTC devices only sporadically report data and as such metrics such as total network data throughput are not relevant, as there is a maximum amount of required transmitted data. In addition, as frame confirmation is not a scalable approach for all frames [31], and blind retransmissions are not scalable [31], the key metric in performance evaluation in this case is packet delivery ratio (PDR). Delay is also measured to ensure the introduced delay by the system never exceeds the mMTC tolerable delay of 10s. In these simulations, the PDR during the initial two runs of the algorithm (first two hours of simulation)

TABLE II
LoRaWAN Timeslots (33 byte transmission)

Data Rate	Timeslot Length	Timeslots in an hour	Max slot change
DR0	1.812s	1986	5
DR1	0.906s	3972	11
DR2	0.453s	7944	22
DR3	0.227s	15888	44
DR4	0.113s	31776	88
DR5	0.057s	63552	176

are not factored in, as a regular LoRaWAN network would not have so many devices initially connecting to the network at the same time.

A. Results

Figure 3 shows the PDR of LoRaWAN with LTS implemented compared to a regular implementation of LoRaWAN, for an increasing number of devices. The figure shows that the proposed algorithm enables a single LoRaWAN gateway to handle the traffic of approximately twice as many devices, while providing the same level of PDR.

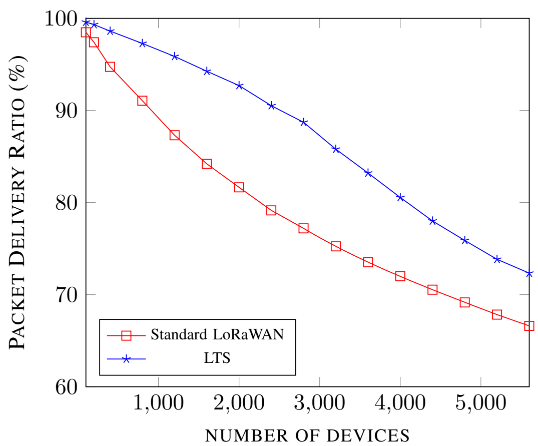

Fig. 3. PDR of a LoRaWAN network with LTS implemented vs standard LoRaWAN, for an increasing number of devices.

A note about the potential scalability of the system: in our simulations we have equally dispersed devices around the gateway in a disk of radius of 4km. However, as well as being more energy efficient, the faster LoRaWAN data rates also enable the splitting of time into smaller timeslots, allowing the handling of approximately twice the number of devices for each faster data rate. In addition, a faster data rate enables a greater $timeslotDelay$ change to be allocated by the Network Server while still maintaining a tolerable delay, providing more flexibility and thus greater potential efficiency of the system. Thus, if devices were dispersed in such a way to enable devices to use a faster data rate on average, there would be increasing gains using our scheme. This could be achieved by deploying a greater density of gateways in the network, while the LoRaWAN Adaptive Data Rate algorithm is enabled on all devices. The number of devices allocated to each data rate in a sample simulation of 5600 devices (equally dispersed) and the resulting average PDR for devices using each data rate, versus the proportion of timeslots available for each data rate is provided in Table III.

Figure 4 shows the mean and max of the delay of transmissions for each data rate, for an increasing number of devices. Note that the mean delay never exceeds 0.06s for $DR4$, $DR5$, and $DR6$. As regular LoRaWAN uses ALOHA without any channel sensing, the delay for regular LoRaWAN is always $0s$. In LTS, the mean delay increases with number of devices, as

TABLE III
LoRaWAN Devices allocated per data rate vs. proportion of timeslots (in a simulation of 5600 devices)

Data Rate	Devices Allocated	PDR	Timeslots in an hour	% of available timeslots
DR0	23.14%	48.34%	1986	1.58%
DR1	21.05%	66.63%	3972	3.17%
DR2	16.32%	79.11%	7944	6.35%
DR3	13.13%	84.53%	15888	12.70%
DR4	11.39%	87.46%	31776	25.40%
DR5	14.96%	88.56%	63552	50.79%

a higher proportion of filled timeslots results in the algorithm having to allocate timeslots to devices farther away from the initial projected transmission time. However, the delayed transmission time for any device never exceeds $10s$.

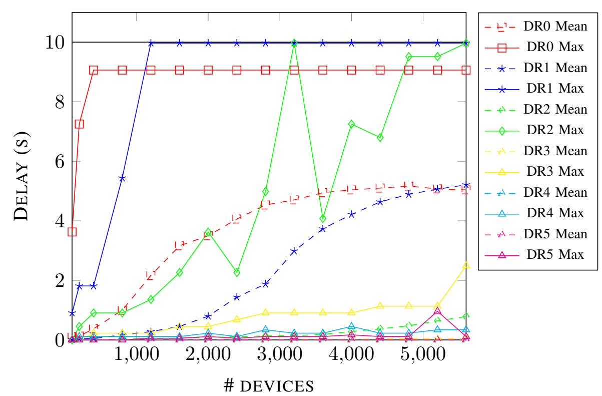

Fig. 4. Mean and max of the delay of an uplink frame of each data rate, for an increasing number of devices

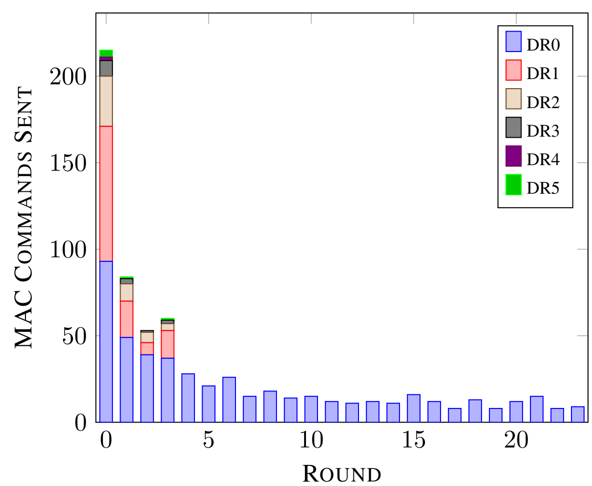

Fig. 5. Number of MAC commands sent after each run of the algorithm, for each data rate (1600 devices)

Figure 5 shows the number of MAC commands that are sent after each run of the algorithm, using each data rate, in an example simulation consisting of 1600 devices. Unsurprisingly, with a roughly equal number of devices allocated each data rate and a transmission using one data rate taking approximately twice the amount of time of the next fastest data rate, the number of collisions increases roughly by two for each slower data rate. The initial run of the algorithm detects and handles the majority of colliding periodicities; the next few runs detect colliding periodicities that were not detected initially because of heavy packet loss in early rounds. The next few rounds also propagate MAC commands that were not able to be sent by the gateways in the initial run because of the duty cycle restrictions of the EU868 band. Eventually for each data rate collisions are avoided and later MAC commands are not required. $DR0$ in this simulation is slow to converge to a situation where no MAC commands are needed per round because the number of devices using $DR0$ is high enough that nearly all of the timeslots are used; however, in a longer simulation the system will eventually converge.

V. Conclusions

In this work, LTS, a novel approach to TDMA over LPWAN has been presented, which is particularly suited to the characteristics of mMTC and LoRaWAN. The scheme does not require extended downlink transmissions from the gateways, and does not require time synchronisation between the devices and the LoRaWAN Network Server. Results from simulations show that the scheme doubles the number of frequently transmitting mMTC devices (once every 10 minutes) that can be handled by a single LoRaWAN gateway while providing the same level of PDR and maintaining a tolerable delay for mMTC use cases, without impacting the ability of the network to send downlink frames or acknowledge high priority packets.

In the future, we intend to extend this work to enable use of different sized packet lengths, and simulate multiple gateways. In addition, we aim to extend the system to perform channel assignment of future uplink frames of devices, rather than keeping the random channel selection of regular LoRaWAN. Given the encouraging results we have found with LoRaWAN, we also plan to investigate the applicability of the approach presented here to mMTC traffic over other non-ALOHA wireless systems. In addition, a proof-of-concept implementation of the system on hardware would strengthen the results shown here. Finally, though the current approach is tolerant of independent event-based traffic because of the periodicity detection method, the approach can be extended to gracefully handle time-correlated event-based traffic through the use of a random-backoff scheme.

Acknowledgment

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

References

[1] Stefan Böcker, Christian Arendt, Pascal Jorke, and Christian Wietfeld. LPWAN in the Context of 5G: Capability of LoRaWAN to Contribute to mMTC. In *IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 737–742, 04 2019.

[2] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub. Toward Massive Machine Type Cellular Communications. *IEEE Wireless Communications*, 24(1):120–128, February 2017.

[3] T. Hoßfeld, F. Metzger, and P. E. Heegaard. Traffic modeling for aggregated periodic IoT data. In *21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 1–8, Feb 2018.

[4] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy. Massive Machine-Type Communications in 5G: physical and MAC-layer solutions. *IEEE Communications Magazine*, 54(9):59–65, Sep. 2016.

[5] E. Soltanmohammadi, K. Ghavami, and M. Naraghi-Pour. A Survey of Traffic Issues in Machine-to-Machine Communications Over LTE. *IEEE Internet of Things Journal*, 3(6):865–884, Dec 2016.

[6] 3GPP. Study on RAN Improvements for Machine-type communications. TR 37.868. Technical report, 2012.

[7] M. Sansoni, G. Ravagnani, D. Zucchetto, C. Pielli, A. Zanella, and K. Mahmood. Comparison of M2M Traffic Models Against Real World Data Sets. In *IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6, Sep. 2018.

[8] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi. Modelling of IoT Traffic and Its Impact on LoRaWAN. In *IEEE Global Communications Conference*, pages 1–6, Dec 2017.

[9] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke. Scalability Analysis of Large-Scale LoRaWAN Networks in NS-3. *IEEE Internet of Things Journal*, 4(6):2186–2198, Dec 2017.

[10] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp. Traffic Models for Machine Type Communications. In *10th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5, Aug 2013.

[11] M. O. Farooq and D. Pesch. Evaluation of Multi-Gateway LoRaWAN with Different Data Traffic Models. In *IEEE 43rd Conference on Local Computer Networks (LCN)*, pages 279–282, Oct 2018.

[12] W. Hsu, Q. Li, X. Han, and C. Huang. A hybrid IoT traffic generator for mobile network performance assessment. In *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 441–445, June 2017.

[13] M. Centenaro and L. Vangelista. A study on M2M traffic and its impact on cellular networks. In *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 154–159, Dec 2015.

[14] Morteza Zarrini and Abdorasoul Ghasemi. Loss and delay analysis of non-Poisson M2M traffic over LTE networks. *Transactions on Emerging Telecommunications Technologies*, 29(2):e3273, 2018. e3273 ett.3273.

[15] P. S. Dester, F. Helder C. dos S. Filho, and P. Cardieri. Performance Analysis of Uplink Traffic for Machine Type Communication in Wireless Sensor Networks. In *IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5, June 2018.

[16] Dae-Young Kim and Seokhoon Kim. Dual-channel medium access control of low power wide area networks considering traffic characteristics in IoE. *Cluster Computing*, 20(3):2375–2384, Sep 2017.

[17] G. Corrales Madueño, Č. Stefanović, and P. Popovski. Reliable and Efficient Access for Alarm-Initiated and Regular M2M Traffic in IEEE 802.11ah Systems. *IEEE Internet of Things Journal*, 3(5):673–682, Oct 2016.

[18] S. Ali, A. Ferdowsi, W. Saad, and N. Rajatheva. Sleeping Multi-Armed Bandits for Fast Uplink Grant Allocation in Machine Type Communications. In *IEEE Globecom Workshops*, pages 1–6, Dec 2018.

[19] K. A. Lizos. Heterogeneous scheduling in wireless networks: Machine to machine case study. In *IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 675–682, Oct 2015.

[20] N. Abuzainab, W. Saad, C. S. Hong, and H. V. Poor. Cognitive Hierarchy Theory for Distributed Resource Allocation in the Internet of Things. *IEEE Transactions on Wireless Communications*, 16(12):7687–7702, Dec 2017.

[21] M. Chen, Y. Miao, X. Jian, X. Wang, and I. Humar. Cognitive-LPWAN: Towards Intelligent Wireless Services in Hybrid Low Power Wide Area Networks. *IEEE Transactions on Green Communications and Networking*, 3(2):409–417, June 2019.

[22] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin. Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling. *IEEE Internet of Things Journal*, 5(3):1830–1842, June 2018.

[23] Roman Trüb and Lothar Thiele. Increasing Throughput and Efficiency of LoRaWAN Class A. In *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, 2018.

[24] Rajeev Piyare, Amy Murphy, Michele Magno, and Luca Benini. On-Demand LoRa: Asynchronous TDMA for Energy Efficient and Low Latency Communication in IoT. *MDPI Sensors*, 18(11):3718, Nov 2018.

[25] K. Abdelfadeel, V. Cionca, and D. Pesch. Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN. In *20th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 02 2018.

[26] Q. Zhou, J. Xing, L. Hou, R. Xu, and K. Zheng. A Novel Rate and Channel Control Scheme Based on Data Extraction Rate for LoRa Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 02 2019.

[27] S. Kim and Y. Yoo. Contention-Aware Adaptive Data Rate for Throughput Optimization in LoRaWAN. In *MDPI Sensors*, 2018.

[28] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda. Evaluating LoRa energy efficiency for adaptive networks: From star to mesh topologies. In *IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, Oct 2017.

[29] Quan Yuan, Jingbo Shang, Xin Cao, Chao Zhang, Xinhe Geng, and Jiawei Han. Detecting Multiple Periods and Periodic Patterns in Event Time Sequences. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 617–626, New York, NY, USA, 2017. ACM.

[30] Titu Andreescu, Dorin Andrica, and Ion Cucurezeanu. *Introduction to Diophantine Equations: a Problem-based Approach*. Birkhauser, 07 2011.

[31] A. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara. Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks? In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2017.