Lightweight Carrier Sensing in LoRa: Implementation and Performance Evaluation

Edward M. Rochester, Asif M. Yousuf, Behnam Ousat and Majid Ghaderi
Department of Computer Science, University of Calgary

Abstract—**In LoRa, leading communication technology for the Internet of Things (IoT), the so-called *Class A* devices are proposed for applications that require low energy consumption. However, the MAC layer of *Class A* devices is based on pure ALOHA, which performs poorly when the network includes a large number of devices. In this paper, we propose a Lightweight Carrier Sensing (LSC) mechanism for LoRa end devices, which does not include back-off and, thus, results in negligible overhead on end devices. LCS is based on the Channel Activity Detection (CAD) procedure already implemented at the hardware level in all modems based on the standard LoRa chipset. We first theoretically analyze the benefits of LCS on a simplified LoRa network to understand its benefits over pure ALOHA. We present the design and implementation of the proposed LCS and provide measurement results to demonstrate its feasibility in real-world LoRa networks. We have also implemented LCS in a detailed custom-build LoRa simulator to study LCS impact on network energy consumption and scalability in large-scale LoRa networks. Our results show that not only LCS supports more end devices, but also results in significantly lower energy consumption compared to ALOHA, thus efficiently improving network scalability without additional complexity or overhead.**

I. Introduction

A. Background and Motivation

The Internet of Things (IoT) refers to a communication paradigm in which everyday objects are connected to the Internet, enabling them to collect and communicate information about their environment. These objects are envisioned to be applied to a vast number of applications, including smart cities and environmental monitoring. The latter two are characterized by extremely dense deployment of devices immersed in the environment [1]. These devices require long-range connectivity but low power consumption. Several Low-Power Wide-Area Network (LPWAN) technologies have been recently developed and standardized to tackle these requirements. These include such technologies as LoRa [2], Sigfox [3], RPMA [4] and Weightless [5], each employing a different technique to achieve long-range low-power operation.

One of the leading LPWAN technologies is LoRa (**Lo**ng **Ra**nge), which is rapidly being deployed around the world. In LoRa networks, a group of devices are connected to the Internet through an access point called a *gateway*. A typical LoRa network potentially has multiple gateways, where each gateway can support thousands of devices [6]. The above is achieved by implementing stateless network access based on the classical pure ALOHA protocol, where devices randomly access the channel and transmit their messages without any prior association or synchronization with the gateways. Such an access mechanism not only allows a gateway to scale to a large number of devices but also leads to simpler end devices, as there is no need for sophisticated signalling between devices and gateways. However, it is well-known that ALOHA-based networks perform poorly under high loads, *i.e.*, when many devices try to transmit during a short time [7]. This could foreseeably occur in high-density Smart City deployments when end devices simultaneously sense the occurrence of some event and attempt to report it to the gateways. Indeed, in [8], [9], it has been shown that the ALOHA protocol employed in LoRa networks is detrimental to the network capacity, as it results in frequent collisions in high-load situations. Although LoRa networks employ several orthogonal channels, in typical deployments, the number of end devices is several orders of magnitude larger than the number of available orthogonal channels. Therefore, while orthogonal channels alleviate the traffic load issue, the problem still exists in each individual channel.

As such, in this work, our goal is to develop and study a light-weight carrier sensing mechanism that can be implemented on LoRa devices with negligible overhead in terms of both processing complexity and energy consumption.

We refer to our mechanism as *Lightweight Carrier Sensing* (LCS) to distinguish it from conventional CSMA. There are two key distinguishing features of the proposed LCS. (i) In LCS, if the channel is found busy, the device *drops* its message. In other words, LCS does not include back-off, whose implementation would add considerable complexity and energy consumption to modestly-capable LoRa end devices. (ii) LCS employs the *Channel Activity Detection* (CAD) mechanism that is already implemented at the hardware level in all standard off-the-shelf LoRa chipsets. Rather than listening to the channel for a fixed sensing time (as in CSMA), the CAD mechanism listens to the channel for approximately one symbol duration, which implies much lower energy consumption.

One may argue that carrier sensing is not effective in wide-area networks due to the hidden terminal problem while consuming precious energy. However, as will be shown in Sections V and VI, the improvement in throughput achieved by the proposed LCS outweigh its energy cost. Intuitively, this is due to couple factors. Firstly, the highest current consumption occurs during data transmission (120 mA assuming 20 dBm transmission power). This means that any packet collision results in a significant waste of energy for all devices involved in the collision (*e.g.*, for the highest spreading factor, this amounts to 1.38 J per device). Thus, preventing even a small fraction of collisions improves the overall energy efficiency of the network significantly. Secondly, the proposed LCS is based on a hardware-based CAD mechanism, which is extremely efficient. It only takes one symbol time to detect channel activity (which can be as small as 1.28 ms), while consuming only 10.8 mA. Even if the CAD mechanism is not very reliable when devices are distributed over a wide area, its impact on end device's energy consumption is meager, when compared with the energy consumed during the actual data transmission.

B. Related Work

In the following, we will briefly review a few recent works that study LoRa network scalability and energy consumption, two key network aspects that are heavily affected by the channel access mechanism.

Network Scalability. An overview of the capabilities and limitations of LoRa is provided in [10]. Using simulations, it is shown that the number of packets successfully transmitted by end devices drastically decreases as end devices transmit packets at high rates. In [11], a stochastic geometry framework is developed to model the performance of a single gateway LoRa network. It is shown that as the number of end devices increases, the coverage probability drops exponentially fast. In [12], it is shown that a LoRa gateway, under perfect synchronization, can support millions of end devices that send a few bytes of data per day. However, under higher traffic loads, with pure ALOHA random channel access, the number of supported devices dwindles to 9 times lower. Using measurements and simulations, scalability of LoRa networks is studied in [6], where it is shown that if the traffic load is very low, a single gateway can support hundreds of thousands of end devices. However, as the traffic load increases, the number of supported devices drops to only several thousand due to packet collisions. As mentioned earlier, it has been shown that the ALOHA channel access mechanism employed in LoRa networks is the leading cause of frequent packet collisions in high-load situations, and hence the main culprit for the limited scalability of these networks [8], [9].

Energy Consumption. To characterize an end device current consumption, lifetime and energy cost of data transmission, analytical models are presented in [13]. The models are based on measurements of currently prevalent LoRa hardware configurations, and physical and MAC layer parameters. Interestingly, the results show that acknowledged transmissions consume less energy than unacknowledged transmissions. In [14], an energy consumption model is developed for LoRa end devices that considers energy consumption by not only processing and transmission but also any attached sensors. The model is used to optimize LoRa configuration parameters, such as spreading factor and communication range, to achieve the lowest possible energy consumption.

The closest two works to ours are [15] and [16]. In [15], a CSMA mechanism with back-off is proposed for LoRa devices. However, as was outlined previously, there are several challenges with implementing MAC layer CSMA in LoRa networks. In [16], Listen Before Talk (LBT) mechanism in LoRa is proposed to sense the channel and, if channel activity is detected, the device backs of the transmission for a random amount of time. The challenge with such an approach is that, similarly to CSMA, implementing back-off will impose additional overhead and complexity on an end device, in addition to the energy consumption that additional wake-up cycles would require.

C. Our Work

In this work, our goal is to develop and study a light-weight carrier sensing mechanism that can be implemented on LoRa devices with negligible overhead in terms of both processing complexity and energy consumption. A key issue in LCS is that performing CAD consumes energy. While ALOHA does not require channel sensing, and thus saves the associated CAD energy, it results in more packet collisions, which not only reduce the throughput of the network but also waste the energy used to transmit the colliding packets. As such, our objective in this work is to provide a systematic study of energy consumption and scalability of LoRa under LCS and ALOHA protocols.

The main contributions of this paper are:

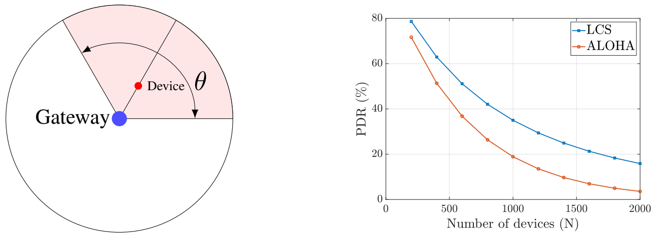
- We depict the benefits of LCS over ALOHA using analysis on a simplified LoRa network model.
- We develop two LoRa energy models to compute the energy consumption of Class A LoRa end devices under LCS and ALOHA.
- We present the design of a simulator to accurately simulate LoRa networks and end devices using configuration parameters measured from our real-world experiments.
- We present and analyze simulation results on energy consumption and scalability of LoRa networks under LCS and ALOHA.
- We present the design of LCS and discuss its operation and implementation on real-world LoRa devices and conduct measurement experiments on a small-scale network.

D. Paper Organization

Our simplified LoRa network model analysis is presented in Section II. Section III describes LoRa energy model. LoRa simulator is presented in Section IV. Simulation results and their analysis are presented in Section V. Section VI presents real-world LCS implementation and testbed measurement results. Section VII, concludes the paper.

II. Packet Delivery Ratio Analysis

To understand the potential performance gains achieved through LCS over ALOHA, we mathematically analyze a simplified network model with LCS capability. We consider a network with one gateway and N devices that are placed uniformly at random in a circular area around it such that all of them are within the coverage of the gateway. The performance metric we consider is the network's Packet Delivery Ratio (PDR), which is defined as the ratio between the number of packets received successfully over the total number of



(a) The network model considered for theoretical analysis.

(b) Theoretical PDR under ALOHA and LCS in a simple network.

Fig. 1: Theoretical network model with resulting PDR under ALOHA and LCS. Parameters used for demonstration: $\theta = 90°$, $T = 20$ minutes, $t_a = 1$s.

packets transmitted. We will assume that, with LCS, a device is capable of detecting transmissions from other devices that are located in a circular sector around it that spans θ degrees of the area. The packet air times are similar among the devices and equal to t_a, and two packets will suffer from a collision if their transmissions overlap in time. We do not consider the capture effect, so all colliding packets will be lost in the event of a collision. Figure 1(a) depicts this network model. We wish to theoretically estimate the PDR in such a network. First, we need to calculate the probability that a device performs a transmission when its packet is ready. We will denote this probability by $P(tx)$. For a generic device whose packet is ready, a transmission will happen if all of the devices within its hearing range are silent, i.e. they have not started a transmission in the past t_a interval. The probability that a device has not started its transmission within the period of time t_a can be calculated as:

$$P(silent) = 1 - P(tx)\frac{t_a}{T}.$$

In order to make calculations feasible, we assume that the transmissions of devices within the θ-sector are independent of each other. Since for large values of N, the number of devices within the hearing range converges to $\frac{\theta}{360}N$, we can write $P(tx)$ as:

$$P(tx) = P(silent)^{\frac{\theta}{360}N} = \left(1 - P(tx)\frac{t_a}{T}\right)^{\frac{\theta}{360}N}. \tag{1}$$

This non-linear equation needs to be solved to find the value of $P(tx)$. In order to compute the PDR, we need to find the probability that a device's transmission is heard without collisions at the gateway. That is to say, all devices in the area are silent during its transmission. Since the devices in the θ-sector can hear the transmission and will remain silent, only the rest of the devices should not start transmitting from t_a seconds before to t_a seconds after the transmission. Therefore:

$$P(rx|tx) = \left(1 - P(tx)\frac{2t_a}{T}\right)^{N - \frac{\theta}{360}N}, \tag{2}$$

where $P(rx|tx)$ is the probability that a device's packet is received successfully if it is transmitted, which aligns with the definition of the network's PDR, so we have:

$$PDR_{LCS} = P(rx|tx). \tag{3}$$

While the PDR under pure ALOHA can be estimated as [7]:

$$PDR_{ALOHA} = e^{-2N\frac{t_a}{T}}. \tag{4}$$

We analyze this model when different number of devices are present in the network. Figure 1(b) shows the PDR for different number of devices in the case of both pure ALOHA and LCS. We can observe that implementing LCS can improve the PDR over ALOHA, specifically, for 1500 device case we can observe 9.8% increase in the PDR .

III. LoRa Energy Model

In this section, we present our models for LoRa end device energy consumption under ALOHA and LCS channel access mechanism. In each case, we compute the current consumption on a device under different LoRa transceiver operations. We assume that the end device uses standard LoRa modem, such as RFM95/96/97/98(W).

A. *ALOHA Current Consumption*

We consider a Class A end device that periodically transmits uplink data packets. After each transmission, the LoRa radio goes into sleep mode. For the next transmission, the radio goes into a wake-up state, then the standby, followed by transmitter initialization and the write of the data to the transmit queue for subsequent transmission. Once the transmission is complete, the device opens two mandatory receive windows (after 1 and 2 seconds) for potential downlink packets (*e.g.*, acknowledgments). Ignoring the radio states with negligible duration (and thus, current consumption), power is mainly consumed when the device is in 'Sleep mode', 'Tx mode' for transmission, or 'Rx mode' for a reception during the receive windows.

As per above, the total current consumption for an unacknowledged packet transmission under ALOHA can be computed as follows:

$$I_{ALOHA} = (T_{TX} \times I_{TX}) + (T_{RX} \times I_{RX}) + (T_{Sleep} \times I_{Sleep}). \tag{5}$$

B. *LCS current consumption*

In LCS, in addition to the current consumption computed in (5) for ALOHA, some additional current is consumed for sensing the channel. In our approach, the CAD functionality implemented in LoRa transceivers is used to check for channel activity. The time taken for CAD is dependent upon the LoRa modulation settings used, *i.e.*, the spreading factor (SF) and the channel bandwidth (BW), and is given by [17],

$$T_{CAD} = \frac{2^{SF} + 32}{BW} \text{ seconds}. \tag{6}$$

During CAD, the end device samples for the presence of a valid LoRa signal and attempts to decode a valid preamble within the recorded samples. As part of this sensing operation, the device also computes the average received signal strength indicator (RSSI) [17]. If a preamble is detected, then the channel is assumed to be busy. Otherwise, we compare the computed RSSI against a threshold specified in the

RFM95/96/97/98(W) datasheet [18] to decide if the channel is busy or idle.

From the above, the total current consumption for an unacknowledged packet transmission under LCS is given by:

$$I_{LCS} = (T_{CAD} \times I_{CAD}) + I_{ALOHA}, \quad (7)$$

where current consumption values in different states are taken from [18] and I_{ALOHA} is calculated as defined in (5). From the datasheets [18] it also can be seen that the most of the current is consumed for data transmission (120 mA), in contrast to the current consumed during channel sensing (10.8 mA).

IV. LoRa Simulator

In this section, we describe the design and implementation of our LoRa simulator, which includes the energy model described earlier.

A. Simulator Design

The simulator is designed as a discrete-event simulator and implemented using Java programming language. It considers all the features of LoRaWAN specification as of version 1.0.2 [2] and is configured to simulate LoRaWAN operations based on North American specifications. The simulator allows for full configuration of gateways, end devices, network parameters and implements ALOHA, as well as LCS for channel access at each device.

B. Packet Reception Model

To determine if a packet is correctly received at a gateway, the gateway calculates the RSSI associated with the packet and compares it with the sensitivity threshold of the LoRa radio receiver used at the gateway [18]. The received RSSI is calculated using the following relation,

$$\text{RSSI} = \text{PX} + \text{GL} + \text{PL}(d), \quad (8)$$

where PX is the transmit power of the end device in dB, GL combines all gains and losses in the transmit/receive path in dB (*e.g.*, antenna gains), and PL(d) represents the path loss in dB assuming that the distance between the end device and gateway is d meters.

C. Wireless Propagation Model

To calculate the path loss, we implemented the log-normal shadowing model, where the parameters of the model are estimated from measurement data presented in [19]. The path loss can be calculated from the following formula:

$$\text{PL}(d) = \text{PL}(d_0) + 10\alpha \log_{10}(d/d_0) + X, \quad (9)$$

where PL(d_0) (in dB) is the reference path loss value at distance d_0 meters from the gateway, α is the path loss exponent, d is the distance between the end device and gateway in meter, and X is the random shadowing modeled as a zero mean log-normal variable with standard deviation σ_x dB. Using measurement data, we considered the reference distance d_0 as 1000 meter. With this information, in the urban environment, we calculated PL(d_0) = 130.12 dB, $\alpha = 2.1$, and $\sigma_x = 7.79$ dB [19].

D. Packet Collision Model

When multiple LoRa transmissions arrive at the gateway at the same time on the same channel with the same BW and SF, several conditions determine whether the gateway can decode one or multiple signals or nothing at all. Following the analysis presented in [20], in our simulator, we determine the collision behavior and capture effect using the following rules. (i) For more than one concurrent receptions at a gateway, if the interfered transmission has non-overlapping preamble and header reception time, and the interferer RSSI is less than or equal to the interfered RSSI, the interfered packet will be received successfully. (ii) If the difference between interferer RSSI and interfered RSSI is greater than 6 dB, the interfered packet will be lost even if the interfered packet has non-overlapping preamble and header reception. (iii) Both interferer and the interfered packet will be lost if there is no non-overlapping preamble and header reception.

V. Simulation Results

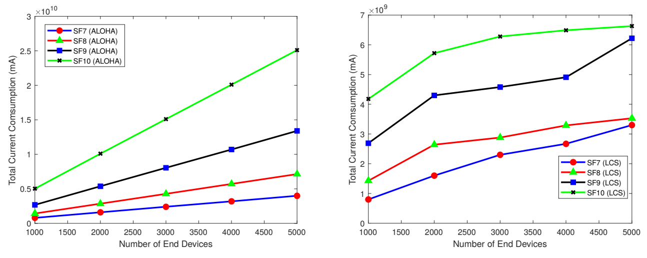
In this section, we present simulation results to compare the performance of LCS and ALOHA in a variety of network scenarios.

TABLE I: Default end device configuration parameters.

Parameter	Value
Tx Power	23 dBm
CR	4/5
BW	125 kHz
Payload size	50 bytes
Packet inter-arrival time	20 minutes

A. Simulation Setup

The default end-device configuration parameters used in our experiments are presented in Table I. The results presented are averaged over 25 simulation runs to ensure that 95% confidence intervals are very small (and hence, not presented on the plots). Each simulation run lasts for 24 hours of simulated time. In the simulations, end devices are placed randomly uniformly in a 500×500 square meters area, where one gateway is placed at the center of the network.



(a) ALOHA current consumption. (b) LCS current consumption.

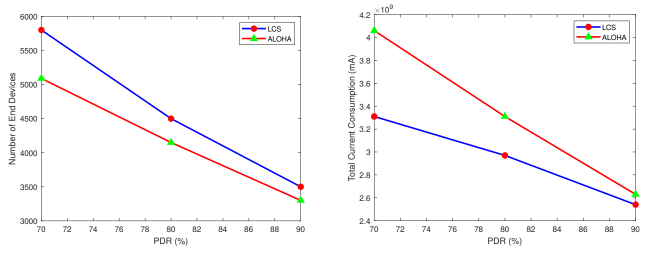
Fig. 2: Energy consumption under ALOHA and LCS.

B. Energy Consumption

The total current consumption for all spreading factors under ALOHA and LCS is presented in Figs. 2(a) and 2(b), respectively. As expected, as the SF increases, so does the current consumption since it takes a longer time to transmit each packet. Also, as the number of end devices increases, for both channel access mechanisms, the total current consumption increases as well, which is expected.

We can also observe that the additional current consumption due to CAD in LCS is negligible compared to the savings achieved by refraining from transmitting potentially colliding packets. Specifically, we see that with ALOHA, the total current consumption increases linearly with the number of devices. However, with LCS, the current consumption has a concave behaviour, *i.e.*, as the number of devices increases, there is a diminishing increase in the total current consumption. The energy savings achieved by LCS over ALOHA are even more significant when considering lower spreading factors. The reason is that, in these cases, it takes much more time and current, to transmit a packet, thus by not transmitting colliding packets, LCS can achieve higher energy savings. Specifically, with 5000 devices, ALOHA consumes 1.2x and 7x more current compared to LCS for SF7 and SF10, respectively.

C. Network Scalability



(a) Max number of supported end devices: LCS supports more end devices at a given PDR.

(b) Total current consumption: LCS results in significantly lower current consumption.

Fig. 3: Scalability under ALOHA and LCS at SF7.

In practice, it is not just energy consumption that matters. Equally important is to achieve some minimum level of throughput to operate effectively. The throughput achieved by a device is directly proportional to the network's PDR. Thus, in this part, we compute the maximum number of devices that can be supported in the network at a given PDR level under ALOHA and LCS, respectively, and then, for this number of devices, we compute the total current consumption.

In Fig. 3 we present the results for the lowest spreading factor, *i.e.*, SF7. We can observe that as the target PDR increases, fewer devices can be supported, resulting in lower current consumption across the network. With fewer number of devices, the difference between the current consumption of both approaches decreases. On top of that, the reduction in current consumption achieved by LCS is more significant as the spreading factor increases.

Specifically, let us focus on the case of 80% PDR requirement. At SF7, the current consumption of ALOHA is 11% more than that of LCS, even though LCS can support 8% more end devices (*i.e.*, 350 more devices). At SF10, we observed that the current consumption of ALOHA is 18% more than that of LCS, while LCS is still able to support 2% more devices (*i.e.*, 20 more devices).

VI. Testbed Experiments

In this section, we present the results of experiments conducted on a small-scale LoRa network consisting of two end devices and a single gateway. The goal is to demonstrate the utility of CAD for detecting concurrent transmissions.

A. Testbed Setup

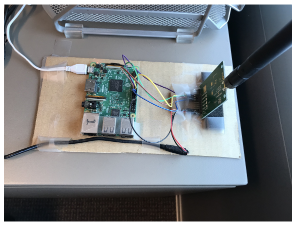
For measuring real-world LCS performance and assessing its feasibility, we have built a small-scale Do-It-Yourself (DIY) LoRa network. The network consists of two end-devices and a gateway, as depicted in Fig. 4. The custom-built gateway operates at the 915 MHz ISM band. The gateway is powered by Raspberry PI 3 Model B, which is connected to a certified 8 channel concentrator board (see Fig. 4(a)) and running open-source multi-channel gateway code [21]. We use the free crowdsourced network server hosted by The Things Network (TTN) [22]. The two end devices consist of Arduino UNO boards connected to RFM95W LoRa transceivers (Fig. 4(b)). The output power for the transceivers was set to 23 dBm. Both devices were set to transmit on the 904.1 MHz channel using the highest possible SF, *i.e.*, 10, at 125 KHz channel bandwidth.

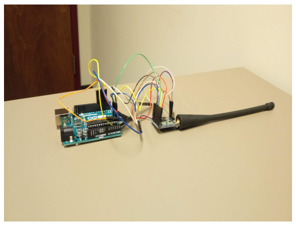
1) LCS Implementation: The end devices run the LMiC library [23], which we modified to add support for LCS by making use of the CAD functionality already built into RFM95W transceiver chips. Fig. 4(d) shows the states that the transceiver goes through on each CAD call. The current version of the LMiC library does not implement the CAD. Thus, we added CadDone and CadDetected mappings into DIO (Digital Input/Output) Mapping, based on the RFM95W datasheet [18]. LMiC library uses polling for the job queue that is updated on every loop run of the microcontroller. As such, we added a method to set the device in CAD mode and schedule an appropriate task in the job queue. When the corresponding task is popped from the queue, the method that sets up the transceiver into CAD mode is called. Following this, the microcontroller moves on to serve other jobs within the queue.

Since the LMiC library only uses a software-defined interrupt handler it limits the accuracy of timekeeping of the device (see further discussion in Section VI-C). In addition to the above, we modified the interrupt handler to use the newly defined pins for CadDone and CadDetected interrupt detection. When only CadDone interrupt is detected, the controller is notified through an event-based system, that no channel activity is detected. This prompts the end device to start the transmission. Otherwise, if any activity is detected, CadDetected interrupt is received, and consequently, the currently pending transmission is dropped. The modified LMiC library with example Arduino code, as well as the custom simulator source code, can be found at TheThingsLab website (http://things.cs.ucalgary.ca/cad.zip).

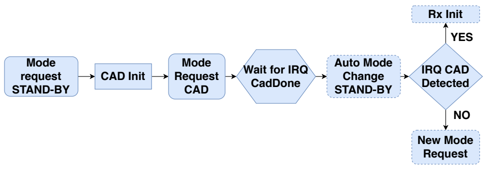
B. Measurement Methodology

Each experiment consists of 1000 packet transmissions, where each message consists of pre-defined LoRa and TTN


(a) Multi-channel gateway.


(b) LoRa end device.

(c) Measurement experiment setup.



(d) LoRa CAD sequence. [17]

Fig. 4: LCS testbed, measurement and operations setup.

headers with hard-coded 5 bytes payload size. The devices are located in the same office space, while the gateway placed in a different room, as can be seen in Fig. 4(c).

C. ALOHA Experiment

We begin by attempting to achieve simultaneous packet transmissions by the two end devices. The expected result is to see no packets received at the gateway due to collisions. As described in [23], the RFM95W transceivers do not support accurate timekeeping themselves, thus leaving it to the Arduino microcontrollers to track time. Combined with the Arduino processor clock drift, synchronous transmission timing differences and delays that arise from the interrupt pooling approach, in our measurements, we saw some packets still being received and decoded on the gateway side. Specifically, out of 1000 simultaneously transmitted packets over 5 repeated experiments, on average, we observed $3-4$ packets received (*i.e.*, we observed the loss of 996-997 packets).

D. LCS Experiment

In CAD mode, the transceiver attempts to capture LoRa preamble symbols from the channel and then search for the correlation between the ideal preamble form and captured samples [18]. If the preamble symbols are not found in the recorded samples, the transmission is not detected.

In our measurements, we observed that out of 1000 transmitted packets over 5 repeated experiments, the end devices were able to successfully, on average, observe the channel activity 997 times, which is highly accurate. For more detailed LCS feasibility measurements over large geographical areas, it is crucial to have a mechanism for consistent end device synchronization on a microsecond scale. One such approach is to use external GPS module timestamps or timers connected to the Arduino microcontrollers. However, this would require rewriting the LMiC interrupt handler implementation to get away from polling and use hardware-based interrupt lines.

VII. Conclusion

In this work, we proposed LCS, a light-weight carrier sensing mechanism for LoRa networks. LCS is based on CAD, which is implemented in LoRa modems, and thus adds little additional overhead or complexity to end devices. We began by analyzing a simplified LoRa network under LCS to depict its benefits over pure ALOHA. We then presented a real-world LoRa network system design that implemented LCS. Finally, we studied the energy consumption as well as the scalability of LCS using detailed simulations. Our result show that not only LCS achieves lower energy consumption, but also allows the network to support more end devices compared to ALOHA.

References

[1] Huawei, "Nb-iot - enabling new business opportunities," Accessed: October, 2019. [Online]. Available: https://www.huawei.com/minisite/4-5g/img/NB-IOT.pdf
[2] LoRa Alliance, Accessed: October, 2019. [Online]. Available: https://www.lora-alliance.org
[3] Sigfox, Accessed: October, 2019. [Online]. Available: https://www.sigfox.com
[4] Ingenue RPMA, Accessed: October, 2019. [Online]. Available: https://www.ingenu.com/technology/rpma
[5] Weightless, Accessed: October, 2019. [Online]. Available: http://www.weightless.org
[6] A. Yousuf *et al.*, "Throughput, coverage and scalability of LoRa LPWAN for internet of things," in *Proc. IEEE/ACM IWQoS*, 2018.
[7] A. S. Tanenbaum, "Computer networks," 2003.
[8] F. V. den Abeele *et al.*, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, 2017.
[9] T. Polonelli *et al.*, "Slotted ALOHA overlay on LoRaWAN: A distributed synchronization approach," *CoRR*, vol. abs/1809.02234, 2018.
[10] F. Adelantado *et al.*, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, 2017.
[11] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Communication Letters*, vol. 6, 2017.
[12] K. Mikhaylov *et al.*, "Analysis of the capacity and scalability of the LoRa wide area network technology," in *Proc. European Wireless*, 2016.
[13] L. Casals Ibáñez *et al.*, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, 2017.
[14] T. Bouguera *et al.*, "Energy consumption model for sensor nodes based on LoRa and LoRaWAN," *Sensors*, vol. 18, no. 7, 2018.
[15] T. To and A. Duda, "Simulation of LoRa in ns-3: Improving LoRa performance with CSMA," in *Proc. IEEE ICC*, 2018.
[16] J. Ortín, M. Cesana, and A. Redondi, "Augmenting LoRaWAN performance with Listen Before Talk," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3113–3128, June 2019.
[17] Semtech, "Reading channel RSSI during a CAD," Accessed: October, 2019. [Online]. Available: https://www.semtech.com/uploads/documents/an1200.21_std.pdf
[18] H. Electronic, Accessed: October, 2019. [Online]. Available: https://www.hoperf.com/modules/lora/RFM95.html
[19] A. Yousuf, E. Rochester, and M. Ghaderi, "A low-cost LoRaWAN testbed for IoT: Implementation and measurements," in *Proc. IEEE WF-IoT*, 2018.
[20] A. Augustin *et al.*, "LoRa scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, 2017.
[21] Lora-Net, Accessed: October, 2019. [Online]. Available: https://github.com/Lora-net
[22] The Things Network, Accessed: October, 2019. [Online]. Available: https:/thethingsnetwork.org
[23] LMIC-Arduino Library, Accessed: October, 2019. [Online]. Available: https://github.com/matthijskooijman/arduino-lmic