LoRadar: LoRa Sensor Network Monitoring through Passive Packet Sniffing

Kwon Nung Choi[1], Harini Kolamunna[1], Akila Uyanwatta[2], Kanchana Thilakarathna[1], Suranga Seneviratne[1], Ralph Holz[1,3], Mahbub Hassan[2], Albert Y. Zomaya[1]
[1]The University of Sydney, [2]The University of New South Wales, [3]The University of Twente
kcho7166@uni.sydney.edu.au

Abstract

IoT deployments targeting different application domains are being unfolded at various administrative levels such as countries, states, corporations, or even individual households. Facilitating data transfers between deployed sensors and back-end cloud services is an important aspect of IoT deployments. These data transfers are usually done using Low Power WAN technologies (LPWANs) that have low power consumption and support longer transmission ranges. LoRa (Long Range) is one such technology that has recently gained significant popularity due to its ease of deployment. In this paper, we present *LoRadar*, a passive packet sniffing framework for LoRa's Medium Access Control (MAC) protocol, LoRaWAN. *LoRadar* is built using commodity hardware. By carrying out passive measurements at a given location, *LoRadar* provides key insights of LoRa deployments such as available LoRa networks, deployed sensors, their make, and transmission patterns. Since LoRa deployments are becoming more pervasive, these information are pivotal in characterizing network performance, comparing different LoRa operators, and in emergencies or tactical operations to quickly assess available sensing infrastructure at a given geographical location. We validate the performance of *LoRadar* in both laboratory and real network settings and conduct a measurement study at eight key locations distributed over a large city-wide geographical area to provide an in-depth analysis of the landscape of commercial IoT deployments. Furthermore, we show the usage of *LoRadar* in improving the network such as potential collision and jamming detection, device localization, as well as spectrum policing to identify devices that violate the daily duty-cycle quota. Our results show that most of the devices transmitting over the SF12 data rate at one of the survey location were violating the network provider's quota.

CCS Concepts

• **Networks** → **Network measurement**; **Sensor networks**; Network reliability; • **Computer systems organization** → **Embedded systems**.

Keywords

IoT, LoRa, LPWAN, Network Traffic Monitoring

1 Introduction

With the growth of Internet of Things (IoT), several IoT-specific radio access protocols have emerged to address the challenging communication requirements and energy constraints of IoT devices. Long range IoT communications with minimum power usage (commonly known as LPWAN protocols) have received particular attention, with LoRa becoming increasingly popular due to its use of an unlicensed frequency band, ease of deployment, low cost, and flexibility in choosing an operator [32]. LoRa sensors are currently used in applications in the likes of smart cities [12, 25], agriculture and livestock management [49], transport and logistics [39], and manufacturing [18]. A total of 133 LoRa operators currently exist globally [4] and many more customer-managed gateways connect to open networks such as The Things Network (TTN).[1] Hence, it is important to develop measurement tools that allow the investigation of performance, utilization, and security aspects of LoRa networks analogous to the tools available for WiFi and cellular networks such as inSSIDer [33], Kismet [55], and ETM770 [41].

LoRa measurement tools are especially needed for three reasons;

Network Performance & Troubleshooting. It is important to monitor performance metrics such as signal strength of sensor communication, level of congestion for frequency bands in use, and average rate of packet loss to diagnose the quality of sensor networks. By analyzing these metrics, network providers are able to compare their relative network performance and obtain a competitive edge in the market by identifying optimal location for maximum gateway communication performance. Parties planning to deploy sensors can uncover marketing opportunities by identifying the number of active devices and their manufacturers to understand the density and type of sensors currently deployed.

Situational Awareness. In case of natural disasters or tactical operations in an unfamiliar territory, it is a common practice to conduct wireless scans to assess what kind of operational wireless infrastructure are present and check whether any type of communication is on-going. Such data gathered can be crucial and provide vital information about survivors or telemetry from a region where the support or tactical teams have limited access [21]. Due to its increasing deployments, we believe LoRaWAN is capable of providing information comparable to what might be collected from other wireless networks such as cellular or WiFi networks.

Research & Development. Analysis of data collected through packet sniffing tools and software have greatly benefited the advances in wireless technologies such as WiFi and Bluetooth. For example, significant amount of security and privacy vulnerabilities in WiFi networks were identified through passive packet sniffing. As an emerging technology, the real-world performance and vulnerabilities of LoRa's MAC layer, LoRaWAN, are yet to be discovered.

Majority of existing LoRa studies are controlled experiments between known gateways. For instance, Blenn and Kuipers conducted a large-scale study using an API provided by a LoRaWAN network provider to obtain the data from all its gateways for a period of six months [7]. Similarly, Yousuf et al. deployed sensors in a city-wide setting to measure throughput, coverage and scalability using their

[1]https://www.thethingsnetwork.org/

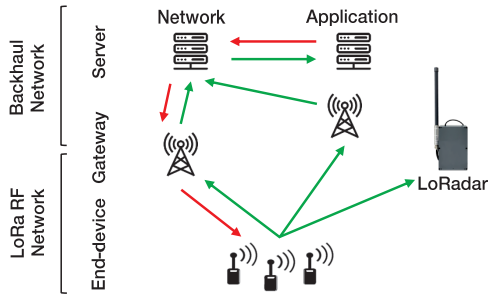


Figure 1: Overview of LoRaWAN communication.

own gateway [57]. In contrast, our focus is to conduct an in-the-wild analysis, and for that we present *LoRadar* which is the first unlicensed measurement tool that leverages the packet logging and decoding of Semtech to present a deeper level of analysis that is user-friendly. Our work is complementary to [19] and addresses its limitations of being able to scan only one-channel at a time which can lead to packet losses on non-listening channels. *LoRadar* uses a commercial grade chip capable of simultaneously scanning 8 channels and obtains data in a passive manner, allowing our work to freely survey key LoRa deployment locations to achieve an in-the-wild study. Also, we provide wider analysis as well as various use cases of *LoRadar* in this paper.

LoRadar is built using commodity hardware and is not bound to a network provider. This allows the discovery of key information across all network providers such as active LoRa sensors, their data transfer patterns, activation methods and the networks they connect to. Next, we carry out a passive network measurement study of LoRaWAN; analogous to early war-driving for WiFi networks [8, 40, 51], and provide insights on early LoRa deployments. *To the best of our knowledge, LoRadar is the first fully fledged open source LoRaWAN scanning tool and our measurement study is the first large scale study to characterize LoRaWAN traffic using passive packet capturing across a multitude of operators.* More specifically, the following are the main contributions of this paper.

- We present *LoRadar*, a software and hardware framework that can be easily setup for passively measuring LoRaWAN networks.
- We validate the operation of *LoRadar* in terms of accuracy and packet loss ratio by conducting controlled experiments. We show that the results are accurate for all packet fields and that there is no significant packet loss in the USB-to-mini PCI-e connection.
- We conduct a large scale measurement study targeting several key IoT deployment sites covering a large geographical area and collect network traffic generated by various IoT sensors connecting to the Internet. Overall, by taking measurements in eight locations over a total of 56 days, we were able to identify 316 unique LoRa sensors and collect 67,704 data frames transmitted from the sensors to the LoRa gateways.
- We extract useful insights without any prior knowledge on commercial networks nor by actively connecting to any of the networks, e.g, network providers operating in a particular region, IoT sensor manufacturer information and their data transfer patterns.
- Using the *LoRadar* measurements, we were able to identify anomalous LoRa behaviors and breaches of spectrum policy. As examples, we find that channel distribution in one of the deployment sites we monitored is poorly managed, with around 90% of transmissions clustered at one data rate. We also find that in another site, around 85% of sensors transmit more often than advised.

The remainder of this paper is organized as follows. In Section 2, we provide a brief background on LoRa protocols and operation. In Section 3, we explain the *LoRadar* hardware and software setup, with experimental validation of the operation of *LoRadar* presented in Section 4. In Section 5, we present real-world LoRa deployment analysis and illustrate various usecases on *LoRadar* in Section 6. Section 7 presents related work and Section 8 concludes the paper.

2 Background

LoRa and LoRaWAN. Long **Ra**nge (LoRa) refers to a physical layer access technology that employs a variant of the chirp spread-spectrum modulation. This modulation is robust to channel noise, multi-path fading and the Doppler effect, even at low power. Therefore, LoRa enables low powered transmission of small data rates (0.3 kbps to 50 kbps) over long distances (up to 15 km in suburban and 2 km in dense urban areas) [27]. Spreading factor (SF) is one of the key parameters that impacts the communication performance of LoRa such as power consumption, range and data rate. SF is defined as the duration of the chirp, which takes values from 7 to 12. A larger SF means longer time on air and increased energy consumption. Next, as larger SF are less sensitive to noise, it can achieve longer communication ranges. The data rate depends on both the bandwidth and SF. Higher bandwidths and lower SF result in high data rates.

LoRaWAN is a star topology network architecture developed on top of the LoRa physical layer for communication between gateways and sensors. It is a Medium Access Control (MAC) protocol designed for transmitting packets and controlling sensors through MAC commands. Fig. 1 depicts the key components of a LoRaWAN setup.

Uplink and Downlink. LoRa sensors broadcast messages as an uplink session which is received by all gateways within the coverage area. The gateways listen to these messages over 8 different channels simultaneously. Online gateways relay the received packets to a network server, where they are decrypted and CRC checked for errors. While offline gateways are usually non-operational and unable to listen to broadcast messages, *LoRadar* is a special application of an offline gateway that listens to the messages without relaying them. Once received, the network server forwards the relayed messages to the application server. Communications between the gateway, the network server, and the application server are usually established over the Internet using TCP/IP, usually over a wired medium.

During a downlink session, the network server encrypts the message and selects a gateway with the highest uplink signal-to-noise ratio (SNR) and received signal strength indicator (RSSI) to transmit the message back to the device. The sensor devices receive downlink messages in three ways: (1) *Class A devices* open two short downlink receive windows after each uplink transmission, (2) *Class B devices* open additional receive windows at scheduled times, (3) *Class C devices* have two receive windows. The second window remains active until the end device needs to transmit a message.

Sessions in LoRaWAN are predominantly uplink, with most sensors not requiring any downlink response from the gateway. This is to save downstream capacity and reduce packet loss that stems from

Field	Description
Time Stamp	UTC time of when the LoRa frame was received
Channel	Intermediate frequency channel receiving the frame
RF Channel	Radio frequency chain receiving the frame
Frequency	The center frequency of the received signal in MHz
CRC Status	The result of the gateway's CRC test on the frame
Modulation	The modulation technique used
Data Rate	Data rate identifier
Code Rate	Error correction code code rate
SNR	Signal to noise ratio of the received packet in dBm
RSSI	Received signal streength in dBm
Size	Number of octets in the received frame
PHY Payload	The frame payload in HEX

Table 1: Description of key fields in a LoRaWAN packet.

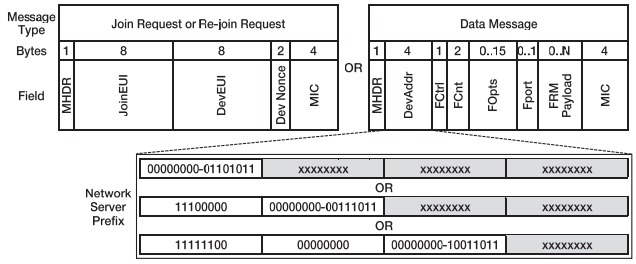


Figure 2: PHY payload structure.

gateways being unable to receive messages when it is transmitting downlink. While downlink sessions provide meaningful insights about sensors, such as identifying Class B devices through the gateway's timed Beacon messages and identifying sensors exhausting the gateway by considering downlink acknowledgment messages, the same information can be obtained through the related bits of the MAC-payload and the uplink acknowledgment messages, respectively. Therefore, monitoring uplink sessions is more effective in understanding LoRaWAN behavior, especially when there is limited hardware to monitor both uplink and downlink simultaneously.

Activation Methods. LoRa IoT devices select between two means of connecting to a LoRaWAN: *Over-the-Air Activation (OTAA)* and *Activation by Personalization (ABP)*. OTAA requires a 64-bit globally-unique device identifier called *DevEUI*, an owner-unique 64 bit identifier called *AppEUI*, and a 128-bit *AppKey* that is obtained from the network operator upon registering the device. OTAA initiates a join procedure prior to transmitting data messages. ABP directly connects devices without a join procedure to a designated network by hardcoding a device address (*DevAddr*), *NwkSKey* and *AppSKey* to the device. Therefore, when an ABP device accesses the network for the first time or after a re-initialization, it transmits the *ResetInd* MAC command in all uplink messages until it receives a *ResetConf* command from the network. OTAA is more secure than APB because new pair of *NwkSKey* and *AppSKey* is generated per session based on two nonces; *DevNonce* and *JoinNonce* residing inside the end device and the gateway respectively. A new pair of nonces is exchanged at the beginning of every session. Validity of these messages are checked through **M**essage **I**ntegrity **C**heck (MIC) with AES encryption using the *AppKey*. Replay attacks are prevented by storing the used nonces at respective places.

Message Types. There are 8 different message types in LoRaWAN. *Join Request*, *Re-join Request*, and *Join Accept* messages are required only by OTAA prior to participating in data exchanges with the Network Server. The uplink channels of *Join Request* and *Re-join Request* messages for the targeted network are determined by the Channel Mask provisioned with OTAA sensors.

The 5 different *Data Messages* (unconfirmed data up/down, confirmed data up/down and proprietary protocol) carry both MAC commands and application data. Confirmed data messages require acknowledgement from the receiver. Proprietary messages are only used by devices that know the corresponding proprietary extensions and follow a different format. Sensors and Network Servers drop unknown proprietary messages.

LoRaWAN Packet Format. A LoRaWAN packet consists of 12 LoRa related fields, as described in Table 1. The *PHY Payload* also contains information regarding device identifiers in Device Address (*DevAddr*), as well as message types within the most significant three bits of the MAC Header (*MHDR*), which is the first Byte within PHY Payload as shown in Fig. 2. As *DevAddr* is provided by the network server, there is a trace of the network operator that can be identified by analyzing the *DevAddr*. As highlighted in Fig. 2, different prefix configurations (1-3 bytes) of the *DevAddr* provide the identification of the connected network servers [37]. The assignment of prefixes to a network operator is done in a way that a 3-bytes prefix is not a subset of 2-bytes prefix and the 2-bytes prefix is not a subset of 1-byte prefix. This is to avoid any confusion when referring to the prefixes to identify the network operator.

3 *LoRadar*

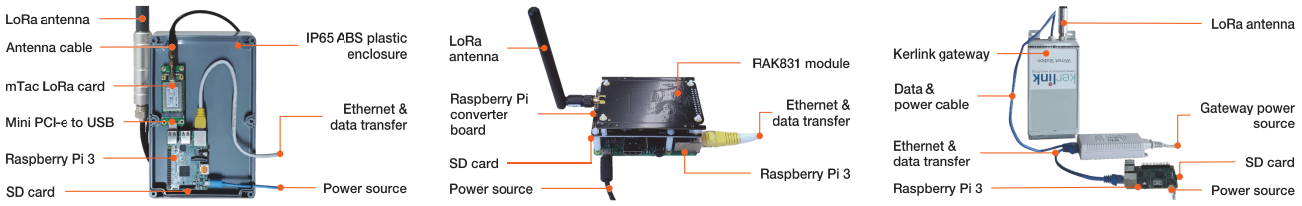
3.1 Overview

Our measurement setup is based on the fact that LoRa sensors broadcast messages and any gateway in the range listening on the same frequency band is able to pick them up. By building an offline LoRaWAN gateway that logs all the messages sent by nearby LoRa sensors and extracting key information, we are able to capture packets sent by LoRa sensors in the neighborhood of *LoRadar*. Indeed, as the data is encrypted with the *AppSKey*, we are unable to read the packet payload. However, we are able to read all the information in the packet header and extract significant amount of wireless link quality related parameters and deployment statistics. This enables understanding the LoRaWAN deployments.

Due to the diversity in LoRa sensor and gateway hardware setups, we believe *LoRadar* must support not only commercial gateways but also custom-built LoRa gateways. Custom-built gateways leveraging single board computers are increasing in popularity due to their low cost (up to 3-times less). Such gateways usually rely on either Serial Peripheral Interface (*SPI*) or USB-to-mini PCI-e connection to operate. To support all variants, we propose three different versions of *LoRadar* as further discussed in the next Section.

3.2 Different *LoRadar* Configurations

We present three different configurations of *LoRadars* to provide support for any commercial and two types of custom-built gateways. As shown in Table 2, Configurations 1 and 2 are custom-built and differ in hardware and connection. Configuration 3 is a commercial gateway that uses our custom logger to collect and extract the data at the gateway level. The components of each configuration are shown in Fig. 3.



(a) *LoRadar* Configuration 1 **(b) *LoRadar* Configuration 2** **(c) *LoRadar* Configuration 3**

Figure 3: Breakdown of components used for *LoRadars*.

No.	Data Rate	Sensor Type	Interval [s]	Activation
1	SF7	Temperature & Humidity [15]	30	ABP
2	SF7	Temperature & Humidity	30	ABP
3	SF7	Dragino GPS Tracker [17]	60	ABP
4	SF7	Dragino GPS Tracker	60	ABP
5	SF8	LoRa GPS Tracker [16]	150	ABP
6	SF9	RN2903 Transmitter [34]	150	ABP
7	SF10	RN2903 Transmitter	300	ABP
8	SF11	RN2903 Transmitter	300	ABP
9	SF12	RN2903 Transmitter	300	ABP
10	SF12	GPS Tracker [14]	1500	OTAA
11	SF12	GPS Tracker	1500	OTAA

Table 3: Summary of LoRa sensors used in validation.

Config.	LoRa Hardware	Connection	Software
1	mTAC LORA [53]	USB-to-mPCIe	Packet Logger
2	RAK831 [46]	SPI	Packet Logger
3	Kerlink Wirnet Station [22]	Ethernet	Python Program

Table 2: Summary of *LoRadar* configurations.

***LoRadar* Configuration 1.** Raspberry Pi 3 [47] was used as the single board computer. We used SX1301 LoRa chip used in mTAC-LORA-915, a gateway accessory card from MultiTech suitable for both 915 MHz and 923 MHz ISM frequency bands, to capture LoRa packets. SX1301 digital baseband chip has the capability of scanning 8 channels simultaneously for preambles of all data rates at all times. Also, the chip is capable of demodulating 8 packets of data simultaneously. Therefore, in *LoRadar*, all 8 uplinks in 923MHz ISM band and all 8 uplinks used by The Things Network in 915MHz ISM band are scanned and demodulated at all times. The demodulation can be done at any data rate in the incoming data packets. A fibreglass 1/2 wave 860-960 MHz antenna with 6 dBi gain was attached to the LoRa card. Connection between Raspberry Pi 3 and the LoRa card was made with a mini PCI-e to USB adaptor. A 32 GB microSD card was inserted for data storage and power was supplied through a micro USB cable. We used the libloragw library [25] for the Raspberry Pi to access the LoRa card and configure radio frequencies. The 922.0-923.4 MHz band-plan setting was selected for 923 MHz, while sub-band 2 was selected for the 915 MHz ISM band based on their popularity among network providers in the country of interest. On top of the library, we collected data through a LoRa packet logger software [26] which records all LoRa packets received by the LoRa card. We selected this based on the convenience of not having to register the gateway on a particular network server and its feature to export data in a csv format. We configured the Raspberry Pi to automatically initiate the packet logger software upon powering on and obtaining the correct time via the Internet.

***LoRadar* Configuration 2.** The hardware setup is near identical to Configuration 1 apart from the following differences. For capturing LoRa packets, we used RAK831, a gateway accessory from RAK wireless that supports both 915 MHz and 923 MHz ISM frequency bands. The gateway accessory is still based on the same SX1301 LoRa chip used in mTAC-LoRa-915, but the connection to the Raspberry Pi 3 is made through SPI. We configured the liblorgaw library to address the different connection setting. The same frequency band-plan setting were used as Version 1 and a 1/2 wave 860-960 MHz antenna with 2 dBi gain was attached to RAK831. Similar to Configuration 1, the LoRa packet logger software was used to collect LoRaWAN packets.

***LoRadar* Configuration 3.** Kerlink Wirnet^{TM} Station 923 MHz gateway was used to capture LoRa packets. This gateway has 915-928 MHz ISM band LongRange^{TM} bidirectional communications capabilities and uses a processor based on ARM 926EJS core for calculations. A fiberglass 1/2 wave 860-960 MHz antenna with 6 dBi gain was connected to the gateway. Gateway was powered over an Ethernet cable and connected to a Raspberry Pi 3 using the Ethernet interface. A 32 GB microSD card was inserted for data storage and powered through a micro USB cable. First, we signed into the gateway using the Raspberry Pi. Next, we scraped data from log files of the gateway and processed using a python program, and generated csv format data output similar to other versions. Gateway logs were saved periodically to a Raspberry Pi to overcome the limited memory capacities of the gateway.

3.3 Data Extraction

Here we explain how information are extracted from LoRa packets.

LoRaWAN Packet Filtering. Since we are only monitoring the uplink traffic, we are able to capture *Join Requests*, *Re-join Requests*, and *Data Messages* that are initiated by LoRa sensors (c.f. Section 2). All fields except for *FRM Payload* in *Data Messages*are not encrypted and can be easily observed through packet sniffing. However, all captured packets using our capturing process may not necessarily be LoRaWAN packets as there may be other applications that use the same ISM frequency band. Thus, we apply a series of rules to filter LoRaWAN packets as described below.

First, we consider region specific configurations to filter the relevant packets [27]. As we are monitoring in two ISM bands (915 MHz and 923 MHz), we follow the LoRaWAN specification [10] to select packets that have; i) 125 kHz bandwidth varying from data rate blocks DR0 to DR5, using coding rate 4/5, and ii) 500 kHz bandwidth at data rate block DR6.

Next, we filter the packets with permitted message types in the *MHDR* (Fig. 2), i.e., *Join Requests*, *Re-join Requests*, and *Data Messages* by applying the following set of rules;

- *Join Requests* and *Re-join Requests* are initiated by OTAA sensors. Only the OTAA activated sensors initiate these messages (c.f. Section 2).
- *Join Requests* and *Re-join Requests* should contain 23 bytes.
- The identified *Data Messages* should contain at least 12 bytes.
- *Join Requests* and *Re-join Requests* should be mapped to a device manufacture and an Application manufacturer (c.f. Section 5.3).
- *Data Messages* should be mapped to a regional network operator (c.f. Section 5.2).

Unique Device Identifiers. Unique device identification process is different depending on the message type.

Join Requests and Re-join Requests. The *DevEUI* is a unique address assigned to each sensor. Hence, sensors that initiate *Join Requests* or *Re-join Requests* can be identified by referring to the *DevEUI* (10-17 bytes of *Join Requests* or *Re-join Requests* packets).

Data Messages. The majority (~86%) of captured traffic contains *Data Messages* from *DevAddr*. *DevAddr* can be duplicated across different *NwkSKey*. Hence, the unique device identification has to be done by considering other parameters. In general, LoRa sensors are programmed to transmit data in defined intervals [20]. Therefore, we also consider the frame count (*FCnt*) and the timestamp of the packets to identify unique sensors. We first cluster packets from the same *DevAddr* per location. Afterwards, packets are further clustered based on the *packets transmission interval* with a padding of two seconds to account for the latency between the transmitted message and obtaining the LoRaWAN packet, based on the maximum time on air of LoRaWAN packets in [3]. We then assign different clusters to different sensors, which provides an approximate number of unique devices.

Transmission intervals. Taking per frame transmission interval as the inter-packet time difference is inadequate as LoRaWAN packets are not guaranteed to successfully reach *LoRadar*, as further discussed in Section 4.1. Therefore, we again consider the frame count differential between successive packets of each sensor and divide the inter-packet time difference by this difference. This means that transmission intervals can only be calculated from the second packet received for each sensor. Consequently, we are unable to provide a transmission interval for sensors that we only receive one packet from and are not used in any analysis requiring accurate transmission interval. We then assign calculated transmission intervals to the corresponding packet.

3.4 Information Visualization and APIs

LoRadar also accompanies a visualization dashboard and APIs for easy data access. The dashboard is built using a Kibana backend and it provides summarized graphs of metadata such as the RSSI and SNR distribution, the proportion of sensor activation methods, and count of unique sensors and observed packets, either globally or location-wise. We also provide support for exporting data in `pcap` format so that packets can be further examined using Wireshark.

Additionally, *LoRadar* provides support for various information extraction requests through its Python script of executable functions. It uses the data file output from the scan and shows the results of the executed function in JSON format. These APIs can be used to obtain information such as a list of transmitting devices, channel-wise data rate usage, RSSI and SNR distribution, visible LoRa networks, and compliance spectrum policy. In Fig. 5 we show the ChannelOccupancy API as an example. As illustrated, the `ChannelOccupancy` API takes a *LoRadar* trace as the input and outputs the percentage of packets observed in each channel for different data rates. The results can also be filtered for specific data rate and frequency plan.

All software versions, data extraction code, APIs, and dashboard tool have been released on https://github.com/loradar/loradar_tool.

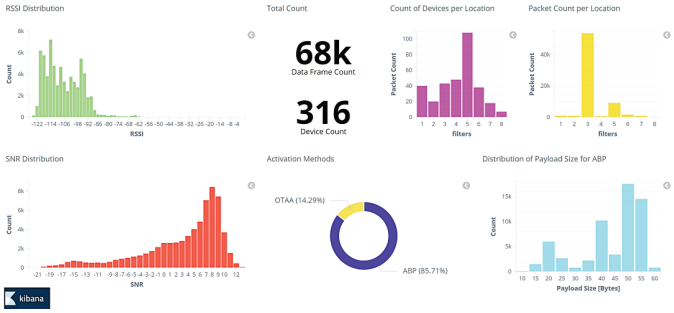


Figure 4: Screenshot of the Kibana dashboard.

Channel Occupancy

```
ChannelOccupancy(input=integer, datarate=integer, freqplan=integer):
    return PercentageOccupancyPerChannel.json
```

Parameters

Name	Type	Required	Description
input	integer	Yes	The data collection session that should be considered for the API. Ranges from 1 to N, where N is the number of collection sessions.
datarate	integer	No	The datarate you want to filter for. Ranges from 7 to 12, referring to SF7 to SF12.
freqplan	integer	No	The frequency plan you want to filter for. Examples include 915, 923.

Figure 5: An example API provided by *LoRadar*.

4 Validation

We validate *LoRadar* in a testbed to show that its information extraction is accurate and that there is no significant packet loss by comparing the measurements with the ground truth. We also validate *LoRadar* in a real network by confirming the statistical information with network administrators.

4.1 Validation in a Testbed

A testbed was created using 11 LoRa end devices—9 ABP and 2 OTAA—as shown in Fig. 6 and summarised in Table 3. The ABP devices were configured to transmit at different data rates, so that the influence of transmission speed on our data collection platform could be investigated. We also configured different transmission intervals ranging from 30 seconds to 1500 seconds to mirror the real-world application of LoRa devices [50, 54].

We used one of the custom-built configurations (Configuration 1) and a commercial configuration (Configuration 3) for comparison since one of our aims is to test whether the USB-to-mini PCI-e connection used in Configuration 1 contributes to packet loss when

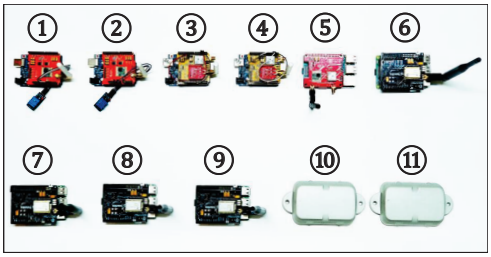


Figure 6: Sensors used for validation with annotation.

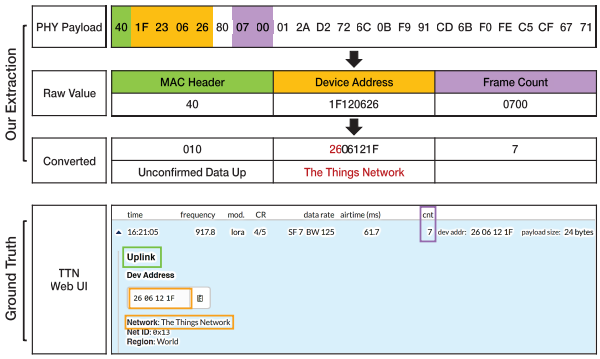


Figure 7: Sample PHY payload from the collected data.

receiving LoRaWAN packets. Previous attempts [36] have shown that USB connection based gateways lead to packet loss during downlink transmissions due to the timing delays associated with USB connections. The gateway transmitted packets at times failed to reach the sensor within the sensor's receive window, causing sensors to re-request the gateway packets. Therefore, we investigate whether timing delays also affect the receiving capabilties of a USB-connection based gateway by using Configuration 1 of *LoRadar*.

First, we investigated whether the information extracted were correct. Information displayed on the TTN web UI was selected as the source for ground-truth as TTN is a reputable free LoRaWAN network provider. We registered one of our sensors with TTN and compared the information presented on the TTN web UI with the information we extracted from the same physical payload. As shown in Fig. 7, our extraction of message type from the MAC Header, network service provider from the Device Address, and Frame Count agree with the information on the TTN web UI for the same LoRa packet.

Second, we compared the data collection capability of the configurations in terms of their proportion of transmitted packets and message types captured, as well as the number of devices identified. We placed both the sensors and the two configurations of *LoRadars* in our lab, as this eliminates other factors such as building obstruction to radio waves from contributing to the data loss.

Fig. 8 shows the number of packets captured by different configurations of *LoRadar* in comparison to the actual number of packets transmitted. The transmitted number of packets were calculated based on the difference between the first and the last frame count seen, as each successive LoRa packet increments its frame count by one. We first investigated whether all of the testbed sensors were identified by matching the list of *DevAddr* and *DevEUI* in the data with those of our testbed sensors. All testbed sensors were identified, and the results are presented by filtering the LoRa packets that were transmitted by these sensors.

Overall, the results show that the USB-to-mini-PCI-e connection used in Configuration 1 does not result in significant packet loss. In fact, Configuration 1 captured more packets than the commercial gateway in some instances. There are uneven numbers of messages received from each device, but this is expected given the chosen transmission intervals where devices 1-4 had significantly lower intervals. Similarly, SF7 is notably higher in count because the same four devices were configured to transmit with a data rate of SF7.

In terms of message types, none of our sensors were configured to require the network to confirm the reception of messages. This is mirrored in the message type figures of Fig. 8, where only *Unconfirmed Data Up* and *Join Requests* are seen. At least 96% of *Join Requests* initiated by OTAA devices were captured by both *LoRadar* configurations in all scenarios. The drop rate of sensors 1 and 2 also contributed to missed *Unconfirmed Data Up* messages.

4.2 Validation in a Real Network

Using configuration 1 of *LoRadar*, we collected data at a central business district (CBD) for a total of 7 days for both the 915 MHz and 923 MHz frequency bands. Based on our data, we identified three different network server providers under 923 MHz and only one network server provider under 915 MHz. To validate our measurement, we contacted the personnel responsible for the LoRaWAN project at the location and learned that sensors were connected to the 923 MHz frequency band of TTN. Initially, 75 LoRa sensors that counted pedestrians and measured temperature and humidity were deployed, with a transmission interval of 900 seconds. However, only half of the sensors were currently active due to running out of power. There were 36 devices in our data that satisfied the characteristics above, and we present the histogram of observed transmission intervals for those devices in Fig. 9. The transmission intervals are greater than 900 seconds in our data due to the latency between the uplink and obtaining the LoRaWAN packet.

5 *LoRadar* Measurements in the Wild

Next, we conducted a state-wide LoRa network performance and situational awareness study by deploying *LoRadar* at a set of geographically distributed key locations. In this section, we show how the extracted information can be used for in-depth analysis of the landscape of LoRa sensor deployments and provide insights on commercially sensitive information leakage in LoRa networks.

5.1 Data Collection Methodology & Dataset

We deployed *LoRadar* at a total of eight locations that were selected based on known LoRa deployment locations listed on reputed online resources such as local newspapers and government announcements, locations with high industrial activity, and other locations to cover a city-wide large geographical area.

In Fig. 10 we provide a geographical map of the measurement locations. To obtain an accurate geographical representation in terms of distance, we first used a mapper API from TTN [29] to identify the locations of TTN gateways nearest to each of the eight locations investigated. The range of RSSI for these gateways was also obtained from the mapper. We divided the range into two groups (≥

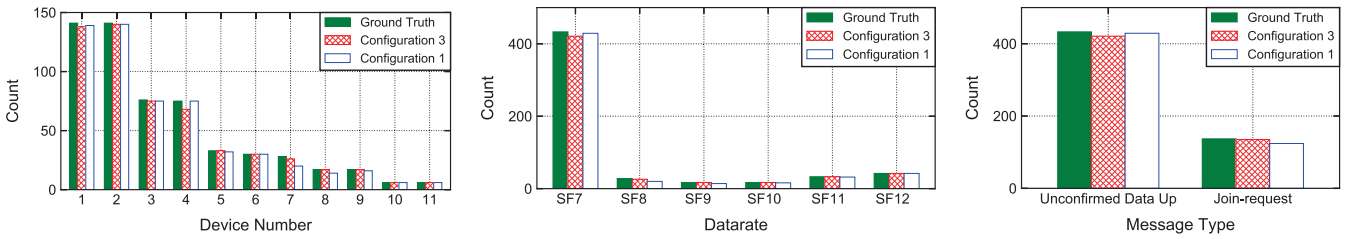


Figure 8: Validation results comparing the packet loss between two configurations of *LoRadar*.

Message Type	915MHz		923MHz	
	Pkts	Dev	Pkts	Dev
Join Requests	281	21	110	24
Re-join Requests	60	1	0	0
Confirmed Data	42905	4	1428	11
Unconfirmed Data	2536	69	20382	186

Table 4: Summary of filtered LoRaWAN packets.

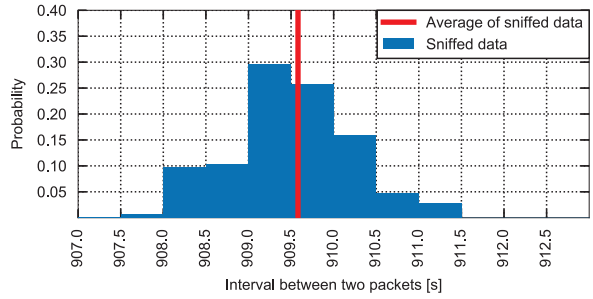


Figure 9: Transmission interval distribution at location 5.

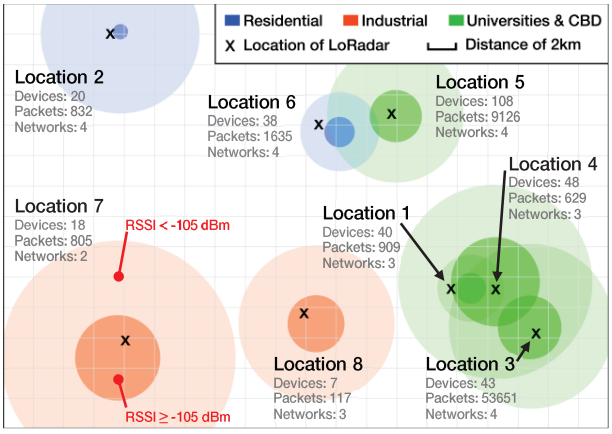


Figure 10: Geographical map of measurement locations.

-105 dBm and < -105 dBm) to represent regions as having good or poor signal strength, respectively. Then, we used Google maps [30] to mark the exact locations of *LoRadar* and the TTN gateways. To obtain a numerical value of the RSSI distance, we measured the Euclidean distance between the location of each gateway and their corresponding range of two RSSI groups. The RSSI ranges are represented by concentric circles, with their Euclidean distances as the radii. Higher RSSI is illustrated by higher color density. Visual information shown on Google Maps was removed for anonymity.

For reliability of data and consistency between locations, both the 915 MHz and 923 MHz gateways were deployed and powered on at each location for 7 days. *LoRadar* was deployed both indoor and outdoor. After each data collection session, gateways were retrieved and their data transferred to a local computer.

Dataset. The summary of filtered LoRaWAN packets of different message types in the two measured frequencies is shown in Table 4. The total number of sensors, LoRaWAN packets, and network server providers are also annotated in Fig. 10. In contrast to the controlled validation experiment, we observed a significant amount of 'CRC_BAD' packets due to very low SNR values. We only considered 'CRC_OK' packets for this study to maintain the accuracy of the results.

Despite the overlap in the RSSI range of some locations (e.g. locations 1 and 4), we did not observe any redundant sensors. Hence none of the location-wise values are a subset of one another. We cluster each location into three categories—residential, industrial, universities & CBD—based on geographical location and advertised information of LoRa deployments.

Highest LoRa activities and number of sensors are observed for Universities and CBD locations. *Location 3* has generated the most number of packets (approximately a few hundreds per hour) whereas the most number of sensors were observed in *Location 5* (108 in total). Upon confirming with the authorities at each of these locations, we know that pedestrian counters based on passive infrared PIR sensors, temperature and humidity sensors are deployed at Location 5. Locations 1, 2, and 3 are utilizing various sensors for research experiments. Location 7 was confirmed to have video-camera based pedestrian counters deployed. Surprisingly, the least number of sensors and LoRa packets were observed in industrial locations.

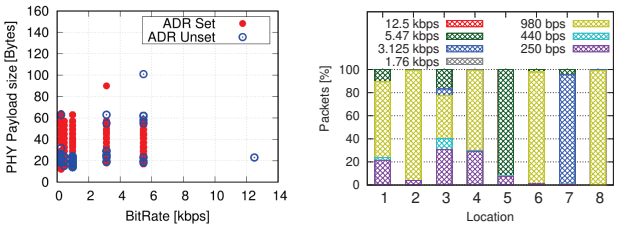
5.2 Network Operators and Sensors

The network operators manage the servers where *Data Messages* are transmitted to. Some providers operate world-wide, e.g., The Things Network and Actility [2]. However, most of the network providers operate only in some specific countries/regions, e.g., *SENET* in USA and *Swisscom* in Europe. Based on the device configurations during the activation process, the network servers assign device addresses to devices. OTAA devices automatically receive a *DevAddr* when joined with *Join Requests*. ABP devices have to request for a *DevAddr* from the network server during the manual activation process.

Fig. 12 shows the total packet counts and the number of observed unique sensors according to the network operator aggregated over all the locations. TTN is the most popular operator for LoRa sensors deployed. Overall, more sensors appear to be deployed on 923 MHz

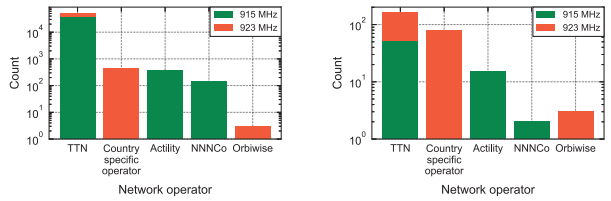
Sensor Manufacturer	Count	Products
Digital Matter Pty Ltd	6	GPS trackers & accelerometers
Multitech Systems Inc.	15	IoT development kits
Microchip Technology Inc	1	General radio modules
Decentlab GmbH	1	Temperature humidity sensors
Turbo Technologies Co.	1	Detectors (smoke & vehicle)
Espressif Inc	4	IoT development kits
Novasonics	1	Unknown
Unidentified (Cravis Co. Ltd)	15	Unknown

Table 5: Sensor Manufacturers.



(a) Payload size vs bit rates **(b) Bit rates in different locations**

Figure 11: Spatial configurations.



(a) Count of packets **(b) Count of unique sensors**

Figure 12: Observed network operators.

band compared to 915MHz band (209 vs 69) whilst more packets are transmitted on 915 MHz (39,792 vs 14,303). There is one operator solely operating on 923MHz band, while TTN is present on both the bands. It has been reported that TTN was initially deployed only on 915MHz band; and started supporting 923MHz more recently due to changing market requirements where more and more sensors started supporting only 923MHz band [11].

Takeaways: Extracted information of LoRadar provides regional statistics of the available network operators in different frequency bands and the number of unique sensors registered with these operators. This knowledge can greatly assist in a tactical operation or an emergency. Moreover, estimation on the number of sensors currently operating in the region for each operator is useful in such scenarios. While the knowledge of the number of sensors and network operators are paramount in certain scenarios, we note that it can be commercially sensitive in Industrial IoT (IIoT) deployments.

5.3 Sensor Manufacturers

As explained in Section 3, *DevEUI* transmitted with *Join Requests* and *Re-join Requests* provides an opportunity to identify sensor manufacturers. Since *DevEUI* is a unique device identifier in IEEE EUI64 address space, it is possible to find the manufacturer with an online API [28]. However, we can do this only if *LoRadar* can capture *Join Requests* or *Re-join Requests* packets which accounts for only 14% of total packets captured in this study. Table 5 lists the sensor manufacturers observed in the areas we investigated. We first observed 15 unidentified devices belong to the same address range. We then further investigated these sensors with *JoinEUI*. Similar to *DevEUI*, *JoinEUI* is also an identifier in IEEE EUI64 address space, which corresponds to the owner of the authentication server for the particular sensor, which resulted in linking these 15 unidentified sensors to Cravis Co. Ltd. Although the knowledge of the manufacturer does not always provide exact sensor type, in most cases it leads to the identification of the type of sensors or the class of sensors as shown in Table 5. For example, the identification of 'Digital Matter Pty Ltd' led us to estimate those six devices are either GPS trackers or accelerometers. Similarly, 'Turbo Technologies Corporation' only manufactures smoke detectors and vehicle detectors. Understanding the type or class of sensors currently operating in a given area provides significant value for situational awareness, especially in emergency response scenarios.

In IoT, this can again be commercially sensitive information. The current OTAA activation method in the LoRaWAN protocol essentially allows us to obtain a reasonably accurate picture of a competitor's IoT deployment.

Takeaways: Sensor manufacturer can be identified from the LoRadar captured Join Requests *and* Re-join Requests *messages initiated from OTAA sensors. This allows the estimation of sensor types, providing a reasonably accurate picture of the IoT deployment.*

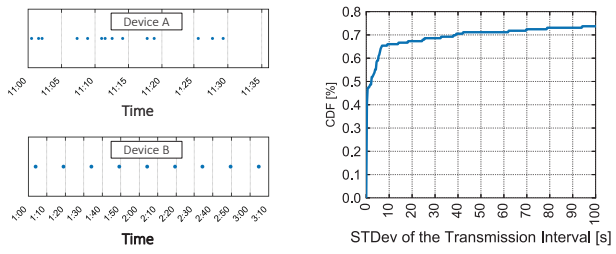
5.4 Wireless Network Configurations

The maximum allowed bit rates are defined by the Data rate, namely the chosen Spreading Factor (SF) and the required bandwidth of the sensor [35]. LoRaWAN allows sensors to dynamically adjust the bit rate through the 'Adaptive Data Rate' (ADR) parameter, setting the most significant bit of the *FCtrl* field of the *Data Messages* (c.f. Fig. 2). Fig. 11a shows the relationship between the maximum allowed bit rates and the size of the PHY payload relative to the ADR status. We observed that 1/3 of the total packets are not using ADR and these packets are using lower payload sizes compared to the packets that are using ADR. On the other hand, the selection of bit rates does not majorly impact on the PHY payload size as expected, because payload sizes are in the range of Bytes.

We then investigate whether there are location-specific configurations of bit rates in Fig. 11b. The observed bit rates significantly vary among locations. This is possibly due to the applications that are being used in these specific locations. For example, *Location 7* with its potential pedestrian counting utilizes a completely different bit rate (3.125kbps) compared to all other locations. However, overall, we observed that 89% of the configurations for the highest two data rates (12.5kbps and 5.47kbps) uses ADR where only 60% of the lower data rates use ADR.

Payload sizes and data rates can potentially be used to make inferences about the actual functionality of the sensors. For example, an attacker can fingerprint the default transmission patterns of a range of sensors and see whether such signatures are available in the captured traffic. In some situations, especially in IIoT applications functionality of deployed sensors can be commercially sensitive.

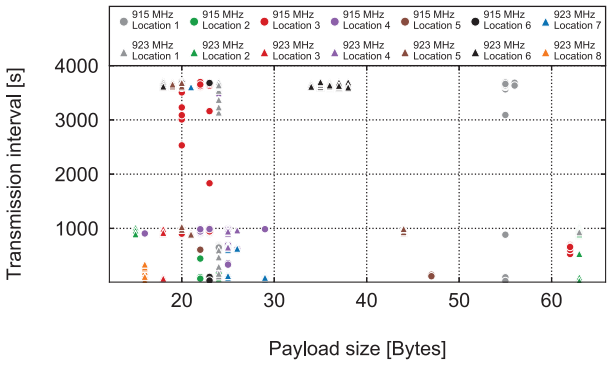
Takeaways: LoRadar provides information on data rates, payload sizes of each device in the scanned area. These can be used to infer the functionality of the sensors.



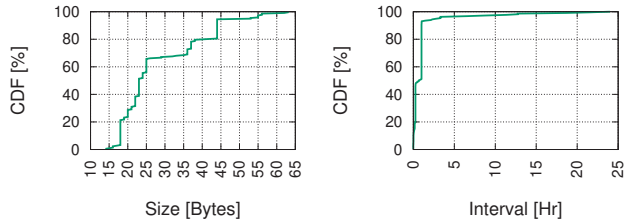
(a) Example devices

(b) CDF of the STDev of the transmission interval

Figure 13: Periodic nature of transmissions.



(a) Payload size vs transmission interval per packet



(b) CDF of payload size

(c) CDF of transmission interval

Figure 14: Transmission interval and packet size.

5.5 Transmission Interval & Size

We analyze the exact transmission time in Fig. 13a with two example devices, which illustrate the temporal behavior of the two types of sensors. For every 10 minutes, *Device B* transmits once periodically whereas *Device A* transmits 4 packets without a regular period. This result shows the potential of fingerprinting sensors through passive traffic monitoring and also the predictive nature of transmissions from certain sensors. Fig. 13b depicts the cumulative distribution of the standard deviation of transmission intervals of all sensors. This shows that over 65% of the sensors are transmitting with intervals that are having less than 10 seconds of standard deviation, further highlighting that next transmission times can be predicted with good accuracy for a larger portion of the devices. Moreover, this also presents a security vulnerability as malicious attackers may exploit this to perform a selective denial of service type attacks [5].

Next, we explore whether there is a relationship of transmission intervals to different devices and deployments. Also in an attempt to isolate different types of sensors by hypothesizing that same payload length may indicate two sensors as the same, we show a scatter plot of transmission interval (in seconds) against the payload size (in bytes) in Fig. 14a. 22% of the total number of sensors were removed from this analysis due to having only one packet and therefore unable to calculate the transmission interval, as explained in Section 3.3. The majority of LoRa sensors appear to transmit within 1000 seconds, with a payload size of 35 Bytes or less. There are two large groups of payload sizes around 25 Bytes and 45 Bytes as shown in Fig. 14b. The horizontal linear patterns reveal 3 different groups of transmission intervals — a long interval of approximately 3,500s, a middle interval of 900s, and a short interval of 10s. We also see that a payload size of around 22 Bytes and an interval of 900s seem to be shared in common by four locations. In general, transmitted payload sizes seem to be different across locations and also clustered into few such as 923MHz location 6, 923MHz location 4, 915MHZ location 3, and 923MHz location 5. Overall, over 90% of sensors have less than 1 hour transmission interval as shown in Fig. 14c.

Similar to other parameters observed, the transmission interval can also be vital information in situational awareness, especially in disaster response. The disaster response team can rely on operational sensors if the transmission frequency is adequate for the operation. It can also be useful in estimating wireless link quality for the purpose of better design new deployments. For example, Location 6 appears to have predominantly long transmission intervals (~3,700s), which suggests lower LoRa wireless network contention despite a large number of sensors around.

Conversely, the ability to predict the new transmission interval for identified sensors opens doors for targeted denial of service or jamming attacks without disrupting the entire network operation.

Takeaways: LoRadar identifies temporal data transmission patterns of each device that may serve in fingerprinting periodically transmitting sensors. This information can be used positively for network troubleshooting or adversely to perform selective attacks. Together with LoRadar provided payload size information, it is possible to isolate the sensors performing the same function.

6 Experimental Evaluation of Further Uses

In addition to providing the statistical information as in Section 5, *LoRadar* provides a deeper analysis of the captured data that helps to improve the network further. Such usecases are explained below.

6.1 Detecting Possible Collisions

6.1.1 Scenario: LoRa messages are more prone to collisions due to their longer air-time. However, there are no collision avoidance mechanisms such as channel sensing and time synchronisation being used in LoRa because its physical layer is ALOHA based. LoRa's concurrent transmissions rely on orthogonal transmissions, i.e, different SF in physically separated channels. However, non-optimized SF and Physical channel selection in large scale LoRa deployments make the collisions paramount. Assume a scenario with a large scale LoRa deployment where the probability of collisions has increased due to increased network density, leading to a high chance of packet loss. In such situations, network administrators require the knowledge of orthogonal transmissions in use, in

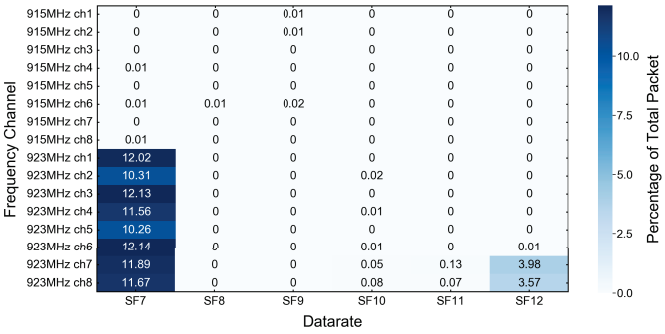


Figure 15: Utilization of orthogonal channels and SF.

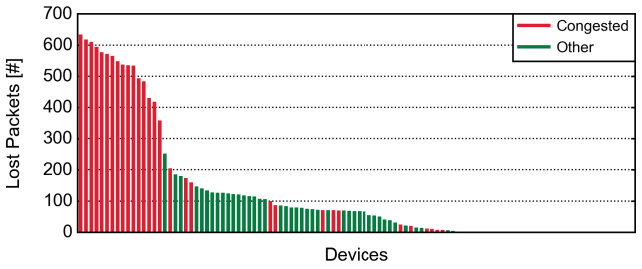


Figure 16: Device packet loss on congested vs other channels.

order to configure the end devices to transmit their data on a less congested channel and SF. However, this is not possible without *LoRadar* as the network servers do not provide overall network statistics in the targeted location that is required to detect possible collisions.

6.1.2 Experiment: We scanned a dense LoRaWAN that is not utilizing the orthogonal channels efficiently. There are nearly 10,000 LoRaWAN packets and around 90% of the packets are transmitted only on 8 orthogonal SF-channel pairs where 96 such transmissions are possible. Then using the `Channel0ccupancy` API in *LoRadar*, we obtained the utilization of different SF and physical channels and present it as a heat map in Fig. 15.

6.1.3 Results: The visualization in Fig. 15 shows that SF7 has the highest congestion for most of the transmissions and few physical channels, with many other orthogonal transmissions being less populated. This transmission configuration will eventually lead to high packet losses. Therefore, we further analyzed the packet loss of each device and their corresponding orthogonal transmissions (See Fig. 16). Most devices that are transmitting in the highly populated physical channel and SF pairs have subjected to the highest packet losses. Eventhough transmissions are in the highly populated physical channel and SF pairs, there may be devices having lower packet loss as the collision has not occurred due to their transmissions not overlapping temporally with other transmissions.

Although there are several work on incorporating channel sense (CS) in LoRaWAN [24, 43, 44], current LoRaWAN specifications does not implement any CS methods. Therefore, inefficiencies caused by these CS methods such as hidden node problem and exposed node problem are not applicable.

6.2 RSSI based device localization

6.2.1 Scenario: Assume a scenario where the location of a malfunctioning device has to be estimated. RSSI based localization method can be used in such scenarios. This method utilizes the propagation loss model (shown in 1) to measure the distances from a node to the beacons, and at least 3 such beacons received at different locations are used in trilateration to obtain the location. However, trilateration cannot be applied to localize the LoRaWAN devices without *LoRadar*because, although the device transmitted data is received by all gateways in the coverage area and sent to the network operator, only the first received data is kept by the network operator and others discarded. Hence, only one RSSI value can be seen in the server console. However, with the use of multiple *LoRadar* placed at different locations, the location of the node can be estimated with the trilateration method.

$$RSSI_d = A - 10n * log(d) \tag{1}$$

6.2.2 Experiment: We placed three *LoRadars* at different known locations and monitored the RSSI values. Then the distance to the node from each *LoRadar* is calculated using the propagation loss model shown in 1. Here, d is the distance, $RSSI_d$ is the RSSI received at the distance d, A is the RSSI received at a distance of 1m, and n is the path loss exponent. The parameters A and n are environment-dependent. In these calculations, we assume that the parameters have consistent values in all directions from the node. To calculate these two parameters, we used the measured RSSI values received from a reference node (R) located at a known location and solved two simultaneous equations. These RSSI values are received by *LoRadar* placed at a distance of d_1 and d_2 from R are $RSSI_1$ and $RSSI_2$ respectively, and the values are calculated using 2 and 3. Then, by substituting these values in 4, the estimated distance d_x from each *LoRadar* x is obtained. Finally, the trilateration method is applied to estimate the location of the node.

$$A = \frac{log10(d_2) * RSSI_1 - log10(d_1) * RSSI_2}{log10(d_2) - log10(d_1)}; \tag{2}$$

$$n = \frac{RSSI_1 - RSSI_2}{10 * (log10(d_2) - log10(d_1))} \tag{3}$$

$$d_x = 10^{(\frac{A - RSSI_x}{10 * n})} \tag{4}$$

6.2.3 Results: Fig. 17 shows the location of the node estimated to a close proximity with around 10 meters of error. This error is due to the assumption of A and n values being consistent. However, in the real world, these parameters have variations due to buildings and other obstacles. Trilateration based localization has many applications and the accuracy can always be improved with more *LoRadar* and reference nodes.

6.3 Radio Jamming Detection

6.3.1 Scenario: Radio jamming is used in wireless data networks such as LoRa to disrupt information flow. Transmission of high power radio signals from jammers decreases the SNR in the targeted frequency of communication. There are three different types of jamming attacks used in LoRa. 1) Continuous jamming - the simplest to execute and involves the jammer to periodically transmit on one channel; 2) Triggered jamming - more sophisticated as it

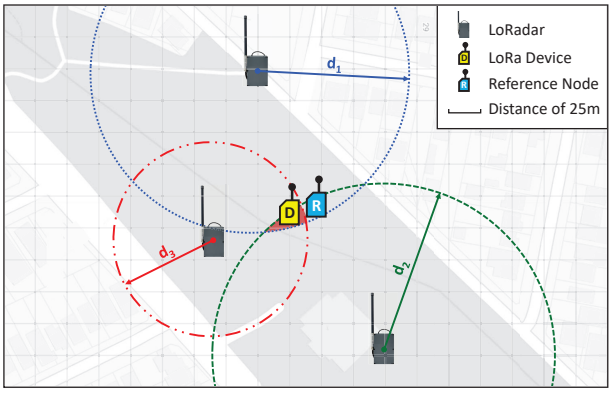


Figure 17: RSSI based device localization.

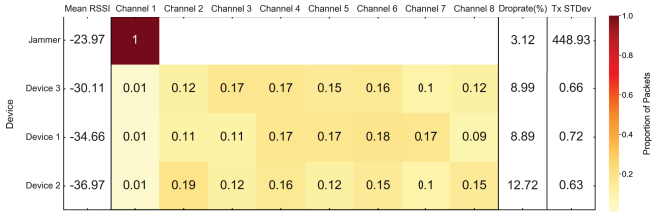


Figure 18: Device mean RSSI, channel-wise packet distribution, droprate and standard deviation of transmission interval (Tx STDev). Devices are in descending order of mean RSSI.

scans a certain channel to detect the presence of any ongoing LoRa messages with the preamble symbols and are triggered to interfere with the transmission; 3) Selective jamming - most complex as it involves listening to part of the LoRa message on any channel and activating only when message headers match those of selected targets (device or message type-specific, or other physical headers).

Assume a scenario where administrators of a certain LoRa network observe a significant increase in drop rates for their devices at a particular region and wish to investigate whether malicious jamming is in action. In such situations, network administrators need information regarding the activity of all devices in the region to identify whether particular devices are transmitting with high RSSI to create deliberate collisions. However, such information is not available from the server-end as network providers are forwarded only the information of devices registered under their network, so malicious parties can assign the jammer a network header that does not match any of the existing networks to hide its packets from the server. *LoRadar* overcomes this limitation by sniffing all LoRa transmissions before they are filtered by the network, making it possible to capture packets from such configured jammers.

6.3.2 Experiment: We created a situation that deliberately interferes with the LoRa data transmission by configuring an off-the-shelf LoRa device to transmit only on one frequency channel with a transmission power of 20 dBm. This was to increase the RSSI differential with other devices that have 14 dBm transmission power similar to regular LoRa devices. We assigned the jammer a network header that did not correspond to any networks in [38]. Based on the required dBm differential between the jammer and the target node calculated in [6] for successive jamming, we configured all devices to transmit on the same data rate to minimize the required dBm differential. We selected the SF12 data rate to increase the collision likelihood, as it has the longest time on air. Each Non-jammer was set with different transmission intervals to simulate real-world deployment with various types of sensors. By syncing the jammer's transmission time with non-jammer devices that were configured to transmit on all channels, we were able to simulate the triggered jamming scenario.

6.3.3 Results: To identify the presence of triggered jamming, we follow four steps:

- *Mean RSSI comparison* - Fig. 18 shows that the mean RSSI of an unknown device (jammer) is higher than those of known devices. Their RSSI differential falls above the required jamming threshold calculated in [6], thus signalling potential packet collision.
- *Channel distribution inspection* - The channel distribution of packets shows that the unknown device transmits only on one channel. Packets from known devices are significantly less on this channel.
- *Drop-rate comparison* - Considerable drop-rate of the known devices shows that the low distribution of Channel 1 is not due to the randomness of channel selection from frequency hopping.
- *High transmission interval standard deviation (Tx STDev) identification* - The high standard deviation of the unknown device coupled with the traits discussed above suggest that the unknown device is a triggered jammer because triggered jammers have highly irregular transmissions due to its dependence on other device transmissions.

A particular device's trace needs to be passed through all four steps for the device to be identified as a jammer. Otherwise, another network's single-channel device or a LoRa device that only transmits upon an event in the close proximity of *LoRadar* will be misidentified as a jammer. While the experiment focuses on triggered jamming, the same method can be used to detect other jamming techniques. Continuous jamming caused by an unknown device is identified by higher mean RSSI that is above the required differential threshold, operate on one channel that also has low utilization by other devices, having low drop-rate than other devices, and have low Tx STDev. Selective jamming is identified in the same way as in triggered jamming, but with the unknown device transmitting on one or more channels.

6.4 Spectrum Policing

6.4.1 Scenario: LoRa sensors also support being manually configured and deployed by users. This creates room for misconfigured LoRa sensors that transmit more frequently than the allowed transmission interval. Such behavior will lead to sensors quickly depleting the daily duty cycle quota placed by the governing bodies, resulting in the loss of data forwarded by the network provider as they cannot enforce duty cycle on uplink messages except preventing sensors from using the network. This will also unnecessarily congest the ISM frequency band. Assume a scenario with a large scale LoRa deployment comprised of different sensor owners. In such situations, radio frequency spectrum managers need to obtain transmission intervals of every device and their chosen data rate. This needs to be compared to the data rate's corresponding

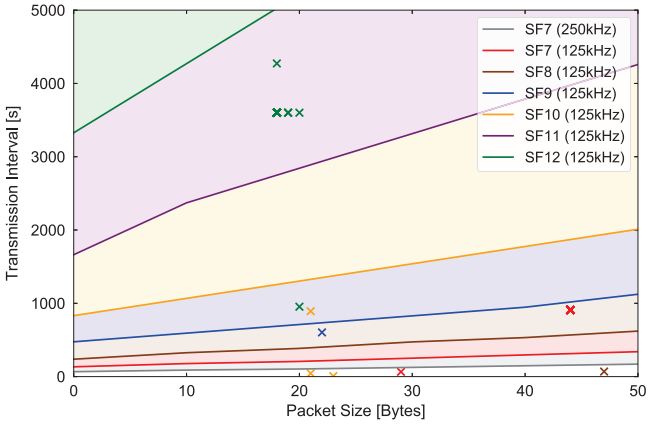


Figure 19: Observed device-wise transmission intervals against regions of satisfactory transmission intervals per data rate. (Best viewed in color)

theoretical transmission intervals for spectral policing. However, obtaining such information from network providers is not plausible as different sensors may be connected to different network providers, but can be addressed with *LoRadar*.

6.4.2 Experiment: We selected one out of the eight LoRa deployment locations we collected from and calculated the transmission interval of each data rate utilized by the sensor. This location hosts a university that is known to conduct various LoRa-related experiments. For the ISM frequency band used by the location, we calculate the lowest-possible transmission interval that will not prematurely exhaust the daily transmission allowance. This calculation is based on the time-on-air (T_{OnAir}) of the transmitted packets (messages), by using the equations provided in the Semtech LoRaWAN modem design guide [48]. We obtain the duty cycle of a device by taking a ratio of its packet-wise time-on-air and the observed transmission interval.

While the duty cycle in most regions is set to 1%, some network service providers employ stricter transmission limitations. TTN, for instance, imposes a Fair Access Policy that limits the uplink airtime to 30 seconds per day per node. To demonstrate the feasibility of *LoRadar* in providing spectrum policing even for such a case, we use equation 5 to calculate the lowest-possible transmission interval within the limitation for a range of LoRaWAN packet sizes obtained at the location. Then using the `SpectrumPolicing` API in *LoRadar*, we obtained a scatter plot of the observed packets.

$$Tx_{min} = \frac{\text{Seconds in a day}}{30} \times T_{OnAir} \tag{5}$$

6.4.3 Results: The visualization in Fig. 19 shows seven different colored regions of suitable transmission intervals for each data rate and packet size. The solid colored lines represent the lowest-possible transmission interval for their corresponding data rates. Colored markers represent the observed transmission frequency of each LoRa sensor and its chosen data rate. Thicker markers denote more number of sensors. Colored markers that are below their respective solid line indicate LoRa sensors that are misconfigured and transmitting more frequently than the recommended interval. Overall, most of the sensors observed at the selected location appear to be misconfigured. For sensors using data rate SF12, all 5 clusters of LoRa sensor configurations fall below its respective green solid line. Such sensors will eventually exhaust their daily duty-cycle quota and lead to network servers ignoring additional messages.

7 Related Work

Empirical evaluations of LoRaWAN in the wild. Relatively few studies obtained empirical data from LoRa networks in the wild.

To the best of our knowledge, none of the work proposed a fully fledged tool and none of the work used only passive sniffing. The most closely related to our work, is the study of The Things Network by Blenn and Kuipers [7]. Authors obtained messages from the API provided by TTN to access the data from all gateways using a particular network session key for a period of six months. On the other hand, some tools have been developed to scan LoRa packets [19]. However, *LoRadar* have more features than these tools such as *LoRadar* can scan all 8 channels simultaneously and provides comprehensive data on the captured transmissions.

Gathering 16.2 million unique frames, they identified a strongly skewed distribution concerning the amount of data transmitted by the sensors. By analysing RSSI and SNR and occasional GPS data in the payloads, authors conclude that the majority of devices are located close to a gateway. Another key observation was that almost all payloads are less than 50 Bytes. Our observations also indicated that payload sizes rarely exceed 32 Bytes, although most are larger than 20 Bytes. Though our own results showed the dominance of TTN in our geographic region as well, we highlight that our work is not limited to TTN only. In fact, our tool is generic and can discover LoRa operators without any prior knowledge.

Yousuf et al. [57] deployed their own LoRa gateways and sensors in a city-wide setting to estimate key performance metrics such as throughput, coverage, and scalability. Our approach is different since we rely only on passive sniffing. Demetri et al. [13] deployed GPS sensors and used TTN Mapper[2] to collect GPS data from end devices and gateways for over a year. Collected data was then used to validate a tool that is used to estimate link quality of LoRa.

LPWAN technology comparison and evaluation. A much larger body of work compared different LPWAN technologies such as LoRa, NB-IoT, and Sigfox, showing that each has application-specific advantages. According to the authors, LoRa and Sigfox outperform others in terms of battery lifetime, capacity, and cost. Nonetheless, NB-IoT offers benefits in terms of data rate, QoS, and range [23, 31, 52]. Casals et al. [9] developed models for the energy efficiency of LoRaWAN devices whilst Brante et al. [20] proposed a multi-antenna setup to enhance the network performance.

Wixted et al. [56] evaluated the range of LoRa end devices with Semtech SX1272 transceivers and Kerlink gateways that was also used by us in the third configuration of our tool. Several work tested the possible range of the technology by conducting controlled experiments. The authors of [42] demonstrated that, with a base station antenna gain of 2 dBi, transmission power of 14 dBm, and configuring the nodes to send packets at SF12, 5km coverage can be achieved on the ground while Radcliffe et al. [45] evaluated the practical range of LoRa networks.

[2]https://ttnmapper.org

In contrast to above work, our proposed software and hardware framework is novel as it is able to passively monitor LoRaWAN without any prior knowledge of devices or network operators, and have the capability of capturing packets in all the channels. To the best of our knowledge, we are the first to conduct in the wild passive measurements of real-world LoRaWAN deployments, study the operational characteristics of live LoRa deployments, and provide insights.

8 Conclusion & Future Work

To summarize, we demonstrated the feasibility of passive packet sniffing in LoRa sensor networks by developing a packet sniffer from off-the-shelf hardware modules and exploiting the fact that any LoRaWAN gateway can listen to all packets transmitted by any sensors in range. We systematically validated the accuracy of information extraction and the robustness of the developed tool; *LoRadar*, by conducting a set of experiments with real devices in controlled settings and in a real LoRa sensor network deployment. We then deployed *LoRadar* in a large geographical region for over 50 days to demonstrate the usefulness of information extracted for network troubleshooting and situational awareness applications. Our measurement results also shed light on possible security vulnerabilities and commercially sensitive information leakage through LoRa networks. We have made the *LoRadar* software available to the research community, enabling anyone to further contribute to improve *LoRadar*. We also believe *LoRadar* will benefit research community to drive further advancements of the technology.

In future work, we first aim to increase the portability of *LoRadar* and enable much wider range of measurement scenarios. We then intend to develop mitigation strategies to limit the sensitive information leakage and predictive nature of transmissions, utilizing *LoRadar* to validate the proposed strategies.

Acknowledgements

This project is partially funded by Data61 and the Defence Science and Technology Group (DSTG) CRP through the Next Generation Technology Fund, and NSW Cyber Security Network.

References

[1] ACM. 2018. Result and Artifact Review and Badging. https://www.acm.org/publications/policies/artifact-review-badging Accessed: 2020-02-13.
[2] Actility. 2019. https://www.actility.com/ Accessed: 2019-10-20.
[3] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine* 55, 9 (2017), 34–40. https://doi.org/10.1109/MCOM.2017.1600613 arXiv:1607.08011
[4] LoRa Alliance. 2019. Coverage and Operator Maps. https://lora-alliance.org/ Accessed: 2020-02-27.
[5] Emekcan Aras, Gowri Sankar Ramachandran, Piers Lawrence, and Danny Hughes. 2017. Exploring the security vulnerabilities of LoRa. In *IEEE Int. Conf. Cybernetics (CYBCONF)*. https://doi.org/10.1109/CYBConf.2017.7985777
[6] Emekcan Aras, Nicolas Small, Gowri Sankar Ramachandran, Stéphane Delbruel, Wouter Joosen, and Danny Hughes. 2017. Selective jamming of LoRaWAN using commodity hardware. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 363–372.
[7] Norbert Blenn and Fernando Kuipers. 2017. LoRaWAN in the Wild: Measurements from The Things Network. (2017). arXiv:1706.03086 http://arxiv.org/abs/1706.03086
[8] Vladimir Bychkovsky, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden. 2006. A measurement study of vehicular internet access using in situ Wi-Fi networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 50–61.
[9] Lluís Casals, Bernat Mir, Rafael Vidal, and Carles Gomez. 2017. Modeling the energy performance of LoRaWAN. *Sensors (Switzerland)* 17, 10 (2017). https://doi.org/10.3390/s17102364
[10] LoRa Alliance Technical Committee. 2017. LoRaWAN 1.1 Specification. (2017).
[11] Core-Electronics. 2019. https://core-electronics.com.au/tutorials/getting-started-on-the-things-network-tutorial.html Accessed: 2019-05-13.
[12] Liverpool City Council. 2019. Smart Pedestrian. https://www.liverpool.nsw.gov.au/business/innovation/smart-pedestrian Accessed: 2019-09-16.
[13] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. 2019. Automated estimation of link quality for LoRa: a remote sensing approach. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 145–156.
[14] Digitalmatter.com. 2019. Compat Long-Life Battery-Powered GPS Asset Tracking. https://www.digitalmatter.com/Devices/3G-GPS-Tracker-Devices/Oyster Accessed: 2019-10-23.
[15] Dragino. 2019. DHT11 Temperature and Humidity Sensor. https://wiki.dragino.com/index.php?title=MyDevices#Example1:_DHT11_Temperature_and_Humidity_Sensor Accessed: 2019-10-23.
[16] Dragino. 2019. LoRa GPS HAT for Raspberry Pi. https://www.dragino.com/products/lora/item/106-lora-gps-hat.html Accessed: 2019-10-23.
[17] Dragino. 2019. LoRa GPS Shield for Arduino. https://www.dragino.com/products/module/item/108-lora-gps-shield.html Accessed: 2019-10-23.
[18] Doug Drinkwater. 2016. Swiss Post Tests IoT LoRa Network in bid to Improve Logistics. https://internetofbusiness.com/swiss-post-to-test-iot-lora-network-to-improve-logistics/ Accessed: 2019-10-16.
[19] Hackster. 2018. Building a LoRa Sniffer with an Adafruit Feather M0. https://www.hackster.io/news/building-a-lora-sniffer-with-an-adafruit-feather-m0-d8b297c7b481 Accessed: 2020-01-10.
[20] Arliones Hoeller, Richard Demo Souza, Onel Alcaraz Lopez, Hirley Alves, Mario de Noronha Neto, and Glauber Brante. 2018. Analysis and Performance Optimization of LoRa Networks With Time and Antenna Diversity. *IEEE Access* 6 (2018), 32820–32829. https://doi.org/10.1109/access.2018.2839064
[21] B. Jalaian, T. Gregory, N. Suri, S. Russell, L. Sadler, and M. Lee. 2018. Evaluating LoRaWAN-based IoT devices for the tactical military environment. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. 124–128. https://doi.org/10.1109/WF-IoT.2018.8355225
[22] Kerlink. 2019. Kerlink Wirnet Station. https://www.kerlink.com/product/wirnet-station/ Accessed: 2019-10-22.
[23] Mads Lauridsen, Huan Nguyen, Benny Vejlgaard, Istvan Z. Kovacs, Preben Mogensen, and Mads Sorensen. 2017. Coverage Comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km Area. *IEEE Vehicular Technology Conference* 2017-June (2017), 2–6. https://doi.org/10.1109/VTCSpring.2017.8108182
[24] Jansen C. Liando, Amalinda Gamage, Agustinus W. Tengourtius, and Mo Li. 2019. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks* 15, 2 (2019). https://doi.org/10.1145/3293534
[25] Libloragw library. 2017. https://github.com/Lora-net/lora_gateway/tree/master/libloragw Accessed: 2019-05-13.
[26] LoRa Packet Logger Library. 2016. https://github.com/Lora-net/lora_gateway/tree/master/util_pkt_logger Accessed: 2019-05-13.
[27] LoRa Alliance Technical Commitee. 2019. *AU915_LoRaWAN_regional_parameters.pdf*. Technical Report.
[28] MACVendors. 2019. Find MAC Address Vendors. Now. https://macvendors.com/
[29] TTN Mapper. 2019. https://ttnmapper.org/colour-radar/?gateway[]=kpmg-iotaa-meshed-barangaroo Accessed: 2019-10-22.
[30] Google Maps. 2019. https://www.google.com/maps/
[31] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. 2018. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express* 5, 1 (2018), 1–7. https://doi.org/10.1016/j.icte.2017.12.005
[32] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. 2019. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express* 5, 1 (2019), 1–7.
[33] Metageek. 2019. inSSIDer. https://www.metageek.com/products/inssider/ Accessed: 2019-10-22.
[34] Microchip. 2019. RN2903. https://www.microchip.com/wwwproducts/en/RN2903 Accessed: 2019-10-23.
[35] The Things Network. 2019. https://www.thethingsnetwork.org/
[36] The Things Network. 2019. Building gateway with MTAC-LORA-H 915. https://www.thethingsnetwork.org/forum/t/building-gateway-with-mtac-lora-h-915/24812/6 Accessed: 2020-06-01.
[37] The Things Network. 2019. The Things Network Device Address Assignment. https://www.thethingsnetwork.org/docs/lorawan/address-space.html#device-address-assignment Accessed: 2019-05-10.
[38] The Things Network. 2020. NetID and DevAddr Prefix Assignments. https://www.thethingsnetwork.org/docs/lorawan/prefix-assignments.html Accessed: 2020-02-07.
[39] IoT Business News. 2019. Tekelek LoRa Tank Monitoring Technology Deployed by Picoty in France. https://iotbusinessnews.com/2019/03/12/80550-tekelek-lora-tank-monitoring-technology-deployed-by-picoty-in-france/ Accessed: 2019-11-01.

[40] J. Ott and D. Kutscher. 2004. Drive-thru Internet: IEEE 802.11b for "automobile" users. In *IEEE INFOCOM 2004*, Vol. 1. 373. https://doi.org/10.1109/INFCOM.2004.1354509
[41] ETM Pacific. 2019. ETM770 - 3g/4g Cellular Network Monitoring Tool. https://etmpacific.com/product/etm770-4g-3g-cellular-signal-strength-meter/ Accessed: 2019-10-22.
[42] Juha Petäjäjärvi, Konstantin Mikhaylov, Antti Roivainen, Tuomo Hänninen, and Marko Pettissalo. 2016. On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology. *2015 14th International Conference on ITS Telecommunications, ITST 2015* (2016), 55–59. https://doi.org/10.1109/ITST.2015.7377400
[43] Congduc Pham. 2018. Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks. *IEEE Wireless Communications and Networking Conference, WCNC* 2018-April (2018), 1–6. https://doi.org/10.1109/WCNC.2018.8376997
[44] Tommaso Polonelli, Davide Brunelli, Achille Marzocchi, and Luca Benini. 2019. Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment. *Sensors* 19, 4 (Feb 2019), 838. https://doi.org/10.3390/s19040838
[45] P. J. Radcliffe, Karina Gomez Chavez, Paul Beckett, and Justin Spangaro. 2017. Usability of LoRaWAN Technology in a Central Business District. In *IEEE 85th Vehicular Technology Conference (VTC Spring)*.
[46] Rakwireless. 2019. RAK831 Gateway Module. https://store.rakwireless.com/products/rak831-gateway-module Accessed: 2019-10-22.
[47] Raspberrypi. 2019. Raspberry Pi 3 Model B. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ Accessed: 2019-10-22.
[48] Semtech. 2013. SX1272/3/6/7/8 LoRa Modem Design Guide. https://www.rs-online.com/designspark/rel-assets/ds-assets/uploads/knowledge-items/application-notes-for-the-internet-of-things/LoRa%20Design%20Guide.pdf
[49] Semtech. 2018. Semtech and Lar.Tech Enable Smart Ranching with LoRa Technology. https://www.semtech.com/company/press/semtech-and-lar.tech-enable-smart-ranching-with-lora-technology Accessed: 2019-09-16.
[50] Semtech. 2019. Semtech's LoRa Technology Locates and Actively Monitors Residents with Dementia. https://www.semtech.com/company/press/semtechs-lora-technology-locates-and-actively-monitors-residents-with-demen Accessed: 2019-11-01.
[51] J. P. Singh, N. Bambos, B. Srinivasan, and D. Clawin. 2002. Wireless LAN performance under varied stress conditions in vehicular traffic scenarios. In *Proceedings IEEE 56th Vehicular Technology Conference*, Vol. 2. 743–747 vol.2. https://doi.org/10.1109/VETECF.2002.1040698
[52] Rashmi Sharan Sinha, Yiqiao Wei, and Seung Hoon Hwang. 2017. A survey on LPWA technology: LoRa and NB-IoT. *ICT Express* 3, 1 (2017), 14–21. https://doi.org/10.1016/j.icte.2017.03.004
[53] Electronic Stocks. 2019. MTAC-LORA-915. https://www.electronic-stocks.com/components/Multi-Tech-Systems,Inc/MTAC-LORA-915.html?gclid=Cj0KCQjw0brtBRDOARIsANMDykZKNZAF4xhLiVv_Q0j-WrPnM2rqSWbuLpaWl744uiBa5rZA0Dvrp4AaAj60EALw_wcB
[54] Ursalink. 2019. Digital Farming Is Creating a More Plentiful, Sustainable Food System (Pilot Project). https://www.ursalink.com/en/success-stories/digital-farming/ Accessed: 2019-11-01.
[55] Kismet Wireless. 2019. https://www.kismetwireless.net Accessed: 2019-10-22.
[56] Andrew J Wixted, Peter Kinnaird, Hadi Larijani, Alan Tait, Ali Ahmadinia, and Niall Strachan. 2016. Evaluation of LoRa and LoRaWAN for wireless sensor networks. In *2016 IEEE SENSORS*. IEEE, 1–3.
[57] Asif M Yousuf, Edward M Rochester, Behnam Ousat, and Majid Ghaderi. 2018. Throughput, Coverage and Scalability of LoRa LPWAN for Internet of Things. *IEEE/ACM International Symposium on Quality of Service* (2018). https://doi.org/10.1109/IWQoS.2018.8624157

9 APPENDIX

We aim to make our entire study reproducible, to allow interested communities to obtain first-hand experience of *LoRadar*'s high utility and further the state of research in LoRa. In this section, we discuss our study in terms of ***repeatability*** (ability of the same team to obtain the same result upon running the same measurement), ***replicability*** (ability of independent teams to replicate our original results upon using our data) and ***reproducibility*** (ability of independent teams to arrive at the same factual conclusion by using their own tools and measurements) [1].

9.1 Repeatability

IoT networks are prone to continuous changes as old sensors get replaced and new sensors are added to the network. The volume and type of sensors deployed, as well as the network operator used highly depend on the location, making strict repeatability challenging. In order to prevent one-time effects, we continuously collected data at each location for at least two weeks, lest a short collection period results in uncaptured packets due to the low frequent transmission of LoRa devices. This can be viewed as 14 repeated daily measurements. We argue that the low standard deviation of transmission intervals in most devices supports the claim that our measurements were repeatable.

9.2 Replicability

To allow others to replicate this paper, we provide all APIs, codes and an anonymized version of the data that conceals sensitive information such as location names. We host these on our Github repository, including documentations on how to replicate our work, accessible through:

https://github.com/loradar/loradar_tool

9.3 Reproducibility

We explain ways to reproduce our results presented in various sections of this paper:

Validation in a Testbed: To reproduce our validation experiment, a team would require some LoRa sensors and have their data rate, transmission interval, activation method configured as outlined in Section 4.1. Each sensor needs to have a valid Device Address assigned and we recommend using The Things Network for this task as it is free.

Once the sensor transmitted packets are collected by the team's own tool, certain sections of the physical payload needs to be extracted. The final three bits of the first byte (MHDR) shows the message type. In terms of device identifiers, ABP devices contain their Device Address in the next four bytes after MHDR, while OTAA devices contain their Device EUI in the next eight bytes after MHDR. The device identifiers are in Little Endian form, and each pair of bytes need to be reversed. Information regarding the network provider and packet frame count are only available for data messages. Network provider prefix is located in the first two bytes of the Device Address, while the frame count occupies the seventh and eight bytes of the physical payload. Frame counts need to be converted from Hexadecimal to Decimal.

Measurements in the Wild: While strict reproduction of our results is difficult due to the reasons explained in Section 9.1, an independent team is still able to obtain the same type of statistics in our study such as the existing network operators, number and types of sensors, wireless network configurations and transmission interval and size. We recommend the data collection duration to be at least three days to ensure sufficient packets are collected from the less frequently transmitting LoRa sensors. The location of data collection should also be as open as possible with good elevation relative to surrounding buildings, to minimize lost packets. The required information extraction steps and querying the obtained information to an online database are explained in Section 5. Furthermore, we provide the exact global configuration file that was used for our in the wild measurement on our Github repository. An independent team may change the key parameters explained in the README file to suit their frequency band.