Aggregation of Contiguous Packets in an Actual LoRaWAN Passive Packet Sniffer

Ahmed Abdelghany, Bernard Uguen, Christophe Moy, Jérôme Le Masson

Aggregation of Contiguous Packets in an Actual LoRaWAN Passive Packet Sniffer

Ahmed Abdelghany, Bernard Uguen, Christophe Moy, *Senior Member, IEEE*, Jérôme Le Masson
Univ Rennes, CNRS, IETR - UMR 6164
F-35000, Rennes, France
{ahmed.abdelghany, bernard.uguen, christophe.moy, jerome.lemasson}@univ-rennes1.fr

***Abstract*—IoT technologies are becoming more pervasive in many critical applications which leads to an increase in the deployment of the popular LoRaWAN (Long Range Wide Area Network) networks. Therefore, it is essential to characterize the network performance and monitor targeted end nodes. In this paper, a passive packet sniffer is proposed that can monitor LoRaWAN at a given geographical location. Using a commercial gateway, a measurement is conducted to passively collect LoRa (Long Range) packets within a duration of one month. From the acquired data, the transmission parameters of the received packets are analyzed. Furthermore, information is extracted from various fields of the frame structure. Then, a Sequence of Contiguous Packets (SCP) is introduced for providing additional packet labeling beyond the Device Address (DevAddr). This original SCP algorithm classifies a stream of packets transmitted by a single end node, based on the similarities between the packets' parameters. The feasibility of the proposed SCP algorithm is confirmed after presenting statistical analysis and the characteristics of estimated SCPs from the real acquired data.**

***Index Terms*—IoT, LPWAN, LoRa, Received Signal Strength Indicator (RSSI), Effective Signal Power (ESP), Packet Sniffing, Smart City, Network Traffic Monitoring.**

I. Introduction

Internet of Things (IoT) has been used in a wide spectrum of domains, such as smart homes, industrial applications and remote monitoring [1]. To address the challenging communication requirements and energy constraints of IoT devices, Low Power Wide Area Network (LPWAN) is introduced. Within this context, LoRaWAN (Long Range Wide Area Network) is currently considered the most efficient technique due to its ease of deployment and operation in the unlicensed frequency bands, i.e. $868\,\mathrm{MHz}$ in Europe and $915\,\mathrm{MHz}$ in the USA [2]. When an end node intends to join the network using the Over-the-Air Activation (OTAA) method, it sends a join request packet containing a 64-bit Device Extended Unique Identifier (DevEUI) for that end node [3]. Then, the network server assigns it a non-unique 32-bit Device Address (DevAddr) to do all the transmissions.

It is important to develop a LoRa monitoring tool capable of acquiring the packet transmission parameters (signal strength, Spreading Factor (SF), frequency, etc.), investigating the level of congestion for frequency bands and SF [4]. Moreover, the number of physical devices and their transmission behavior could be estimated at a given geographical location. This analysis could help the network operator to troubleshoot, plan and evaluate the performance of their network. Otherwise, in case of emergency or for administrative control, it is essential to scan all the active end nodes as well as keeping the track of the targeted devices is vital information in special security operations. On the other hand, data analysis through packet sniffing have a great impact on advancing the Research & Development of the LoRa protocol and identifying many security vulnerabilities.

Recently, some works investigated empirical data from the LoRaWAN network. For example, in [5] the authors perform an exploration of a dataset composed of real LoRa packets, i.e. taken from [6]. Thus, the transmission characteristics, network behavior and message exchanges are analyzed using their software tools. While in [7], a passive packet sniffing framework for LoRa's Medium Access Control (MAC) protocol is introduced. However, these previous works do not solve the problem of classifying the packets which are originated from the same end node while having a non-unique DevAddr. Since DevAddr is dynamic, different end nodes can have identical DevAddr which composes different chains of analogous packets.

In this paper, a passive packet sniffer is proposed using a commercial LoRa gateway and commodity tools. Within the area of the Campus Beaulieu in Rennes, France, the network activity and the transmission parameters of the packets received during one month are statistically investigated. For classifying the packets transmitted by the same end node, a Sequence of Contiguous Packets (SCP) is proposed as a device sub-identifier beyond the DevAddr. Hence, the introduced SCP algorithm properly identifies the sequences of analogous packets with the same DevAddr, using specific criteria. Based on that, the packets, i.e. originated from a unique device, can then be tracked and analyzed, as envisioned in Figure 5 and explained in the upcoming sections.

The remainder of this document is organized as follows. Section II gives an overview of the system setup. Section III provides analysis of the acquired dataset. The SCP is then introduced in Section IV. In Section V, the experimental re-

sults of the proposed algorithms are presented and commented. Finally, the work is concluded in Section VI.

II. Proposed Passive Packet Sniffer

A. Overview and System Configurations

LoRaWAN is a star network topology architecture, thus, an end node broadcast an uplink message which is received by all gateways within the coverage area. Based on that fact, the proposed experiment utilizes a gateway that passively logs all the LoRa packets sent by nearby end nodes, as shown in Figure 1.

A Tektelic KONA Macro Gateway is used whose antenna is fixed on the roof of the university building [8]. Here, the ChirpStack Gateway Bridge service, i.e. part of the ChirpStack open-source LoRaWAN Network Server stack [9], is configured on the gateway to publish the packets' logs to an MQTT (Message Queuing Telemetry Transport) broker [10]. On the other hand, a desktop computer is used that has Node-RED software installed on it [11]. Hence, a Node-RED flow is implemented that starts by subscribing to the aforementioned MQTT broker, and then the packets are streamed to a Python script. This Python script uses the Lora-Packet decoder library, i.e. introduced in [12], for decoding the LoRa physical payload. Then, the packet is timestamped and appended to a data file for further analysis, as explained in the following sections. Furthermore, these data are provided to the research community on this online repository [13].

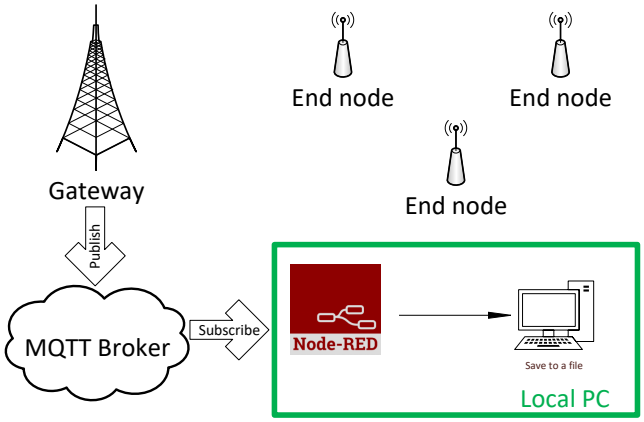


Fig. 1: Setup of the proposed packet sniffer.

B. Data Extraction

The transmission parameters are given by the gateway for each received packet $\mathbf{m}[n]$, where n is the packet index. Beside, decoding of the Lora physical payload gives key information from the frame header about the device identifiers, message type, packet counter, etc. Accordingly, Table I summarizes the main acquired parameters.

TABLE I: Description of key fields for a received packet $\mathbf{m}[n]$

Field	Type	Description
$t[n]$	float	Time at which the packet is received by the gateway
$f[n]$	float	The center frequency of the received packet
$SF[n]$	int	The Spreading Factor of the received packet
$RSSI[n]$	int	The Received Signal Strength Indicator of the received packet
$SNR[n]$	float	The Signal-to-Noise Ratio of the received packet
$ESP[n]$	float	The Effective Signal Power of the received packet
$a[n]$	str	The Device Address (DevAddr) of the received packet
$FCnt[n]$	int	Counter of the transmitted packets, which corresponds to the total number of transmitted packets (by a node) from the beginning until the received packet
$MT[n]$	str	The Message Type (Unconfirmed/Confirmed message) of the received packet
$ADR[n]$	bool	The Adaptive Data Rate bit of the received packet
$PS[n]$	int	The Payload Size of the received packet
$AT[n]$	float	The Air Time of the received packet

III. Dataset Analysis

A. Transmission Parameters

The received data are initially investigated concerning its signal strength. Accordingly, ESP is manually calculated for each received packet as:

$$ESP_{dBm} = RSSI_{dBm} + SNR_{dB} - 10\log_{10}(1 + 10^{\frac{SNR_{dB}}{10}}). \quad (1)$$

As shown in Figure 2, one can observe that the RSSI values saturate when approaching $-120\,\mathrm{dBm}$ due to the noise floor limitation, i.e. $SNR = 0\,\mathrm{dB}$, as proven in [14]. On contrary, ESP breaks this limitation by its enlarged range with even most of its values in the lower region. This recommends utilizing the ESP rather than RSSI in many prospective IoT applications.

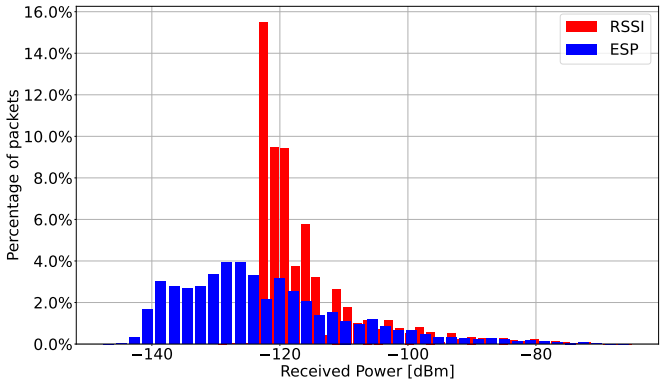


Fig. 2: Comparison of the histogram of RSSI versus ESP.

On the other hand, $AT[n]$ is subjected to the value of the payload size and SF, as mentioned in [15]. Hence, the airtime values of the received packets are computed and compared to the payload size. As shown in Figure 3, it is observed that payload size is limited depending on the SF value. For example, SF7 can theoretically go up to a payload size of 222 bytes. Moreover, SF9 has payload size with a maximum

limitation of 115 bytes which is confirmed by the shown results. Notice that, only one packet with SF12 has been observed above the theoretical limit of 51 bytes.

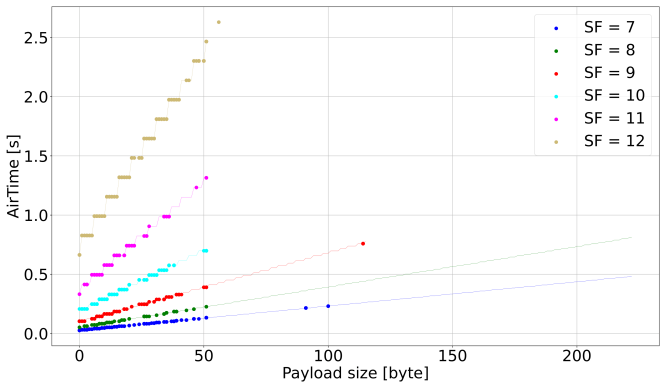


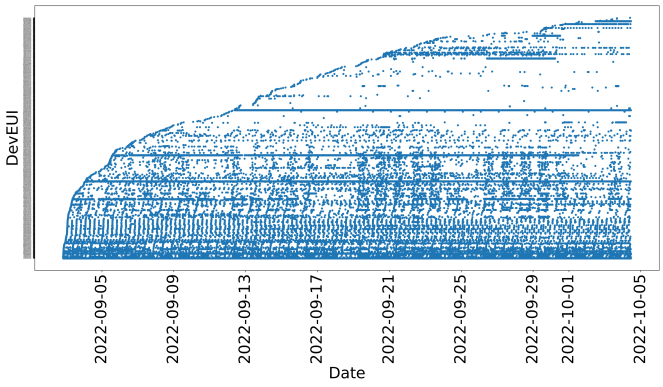
Fig. 3: Measured AT for different SF and payload size values of the actual packets.

B. Device Identifiers

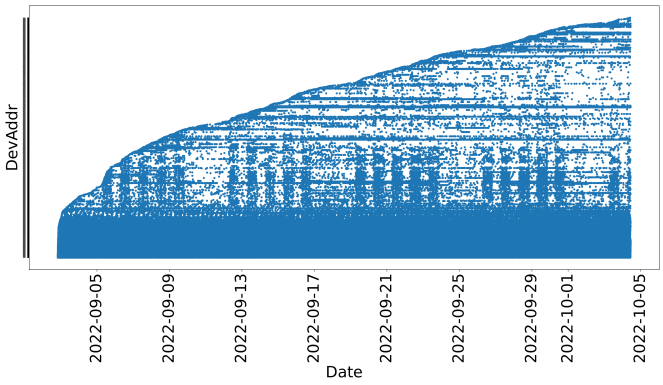
As shown in Figure 4a, the y-axis is composed of the distinct DevEUIs which are obtained from the measurement. Most of these DevEUIs send many join requests within one month. While in Figure 4b, there is a pattern of dense regions that are observed to follow the distribution of the join requests in Figure 4a. In fact, this behavior is reasonable as every join request packet should be followed by a data packet containing the newly assigned DevAddr.

Based on IEEE Standards Association [16], DevEUI is extended from the Organizationally Unique Identifier (OUI) of the constructor or Company ID (CID). Therefore, it is possible to know the manufacturer of the device from the DevEUI. Taking the measured data in Rennes as an example, Table II shows that 29.5% of the detected devices are registered by the IEEE Registration Authority, while 26.5% of them have unregistered OUI. Each of the other observed devices has different manufacturers which can reveal a wide spectrum of sensor types and applications, such as GPS trackers, accelerometers, smoke detectors, etc.

On the other hand, DevAddr is extended from the seven bits of the Network IDentifier (NwkID), while the 25 last bits are randomly chosen [17]. Therefore, the network operator can be recognized for each distinct DevAddr. As shown in Table III, 8.71% of the DevAddr are unassigned to any network, while 21.33% are considered private or experimental projects without an official network but a dedicated range of addresses. Moreover, one of the French network operators utilizes 53.67% of the traffic alone.



(a) The received join request packets against each DevEUI.



(b) The received data packets against each DevAddr.

Fig. 4: The received packets of join request and data types.

TABLE II: Device Manufacturer according to the DevEUI

Organization	Count	Percentage
IEEE Registration Authority US	147	29.5%
OUI_not_registered	132	26.5%
HOMERIDER SYSTEMS FR	93	18.7%
Abeeway FR	64	12.9%
Invoxia FR	15	3.0%
ADEUNIS RF FR	8	1.6%
Microchip Technology Inc. US	7	1.4%
sofrel FR	6	1.2%
Robert Bosch GmbH DE	4	0.8%
GLOBALSAT TECHNOLOGY CORPORATION TW	4	0.8%
XEROX CORPORATION US	4	0.8%
TELEMECANIQUE ELECTRIQUE FR	3	0.6%
PYRESCOM FR	2	0.4%
STMicroelectronics SRL GB	2	0.4%
Itron Inc. US	2	0.4%
Shenzhen RAKwireless Technology Co.,Ltd. CN	1	0.2%
Helium Systems, Inc US	1	0.2%
VG LABORATORY SYSTEMS LTD GB	1	0.2%
Dragino Technology Co., Limited CN	1	0.2%
BH TECHNOLOGIES FR	1	0.2%
Total	**498**	**100%**

TABLE III: Network Operator according to the DevAddr

Operator	Count	Percentage
Bouygues Telecom World	1127	53.67%
Private/experimental nodes Local	448	21.33%
Orange World	237	11.29%
unassigned	183	8.71%
Proximus Europe	13	0.62%
Actility World	9	0.43%
The Things Network World	8	0.38%
SoftBank World	6	0.29%
Charter Communicaton USA	6	0.29%
EveryNet Russia	5	0.24%
Loriot World	5	0.24%
Amazon World	4	0.19%
Axatel Italy	4	0.19%
Cisco Systems World	4	0.19%
Comcast World	4	0.19%
MultiTech Systems World	4	0.19%
Kerlink World	4	0.19%
Ventia World	4	0.19%
NNNCo World	4	0.19%
Netze BW GmbH World	4	0.19%
Shenzhen Tencent Computer Systems Company Limited China	4	0.19%
Swisscom World	4	0.19%
Tektelic World	4	0.19%
A2A Smart City World	4	0.19%
KPN Europe	1	0.05%
Total	**2142**	**100%**

IV. Modelling of Proposed SCP

The dataset $\mathcal{D}$ is composed of a stream of packets as follows:

$$\mathcal{D} = \left\{\mathbf{m}[0], \ldots, \mathbf{m}[n], \ldots, \mathbf{m}[N-1]\right\}, \tag{2}$$

whereas each packet $\mathbf{m}[n]$ belongs to a specific DevAddr a, as depicted in Figure 5. Accordingly, a dataset $\mathcal{D}_a$ is the stream of packets assigned to DevAddr a, this can be defined as:

$$\mathcal{D}_a = \left\{\mathbf{m}_a[0], \ldots, \mathbf{m}_a[i], \ldots, \mathbf{m}_a[N_a-1]\right\}. \tag{3}$$

Moreover, a dataset $\mathcal{D}_a$ can also be defined as a set of SCPs:

$$\mathcal{D}_a = \bigcup_{l=0}^{L_a-1} \mathcal{P}_a^l, \tag{4}$$

where L_a is the total number of SCPs in a given DevAddr a. Furthermore, an SCP $\mathcal{P}_a^l$ is a consecutive sequence of packets which is defined as:

$$\mathcal{P}_a^l = \left\{\mathbf{m}_a^l[0], \ldots, \mathbf{m}_a^l[j], \ldots, \mathbf{m}_a^l[J_a^l-1]\right\}, \tag{5}$$

where a packet $\mathbf{m}_a^l[j]$ is assigned to DevAddr a and belongs to SCP index l. Accordingly,

$$\sum_{a\in\mathcal{S}_A} \sum_{l=0}^{L_a-1} J_a^l = N, \tag{6}$$

whereas $\mathcal{S}_A$ is the set of DevAddrs extracted from the whole dataset $\mathcal{D}$ as:

$$\mathcal{S}_A = DevAddr(\mathcal{D}). \tag{7}$$

To form this SCP $\mathcal{P}_a^l$ across a given dataset $\mathcal{D}_a$, sequential packets are concatenated based on checking the comparable parameters of each two consecutive packets. As stated in Algorithm 1, not having a negative packet counter difference $FCnt_{diff}$ while being assigned with the same DevAddr a is the initial solid condition. In case of packet losses between two packets whose $FCnt$ values are relatively low, the concatenation is avoided with condition C_1. For example, if the minimum value of Packet Delivery Rate (PDR) is set to 0.75 while $FCnt[i] = 1$, only one packet loss is allowed. In fact, condition C_1 tries to avert the uncertainty of the concatenation process while having a sequence of recurrent values of $FCnt$, as very often happens with low values of $FCnt$. Otherwise, a limited amount of lost packets is allowed, e.g., 20 packets in the presented algorithm, which could be adjusted depending on the application requirements. Accordingly, additional conditions are checked to increase the robustness of the algorithm, such as the two appended packets must have the same message type and ADR bit. Besides, they must either have the same SF or payload size.

For a second aggregation phase, the same algorithm is applied to link the discontinuous SCP in each DevAddr, as shown by the two overlapped SCPs which are colored by red and blue in Figure 5. This overlapping scenario often happens while using the Activation by Personalization (ABP) method with a hardcoded DevAddr. Therefore, the probability of having an identical DevAddr assigned to more than two devices at the same time increases. For that, the same conditions, i.e. stated in Algorithm 1, can be used to check the comparable parameters between the last packet $\mathbf{m}_a^0[J_a^0-1]$ of SCP $\mathcal{P}_a^0$ to the first packets $\{\mathbf{m}_a^1[0], \ldots, \mathbf{m}_a^{L_a-1}[0]\}$ of the successive SCPs $\{\mathcal{P}_a^1, \ldots, \mathcal{P}_a^{L_a-1}\}$. If the conditions are $True$ for many consecutive SCPs, only the closest analogous SCP is assigned to the initial SCP $\mathcal{P}_a^0$. This process is repeated for all the SCPs of the dataset $\mathcal{D}_a$, then, the connected SCPs are detected using a graph structure [18] to be aggregated.

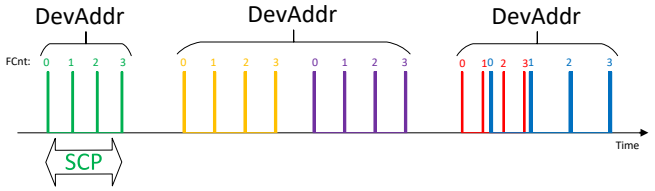


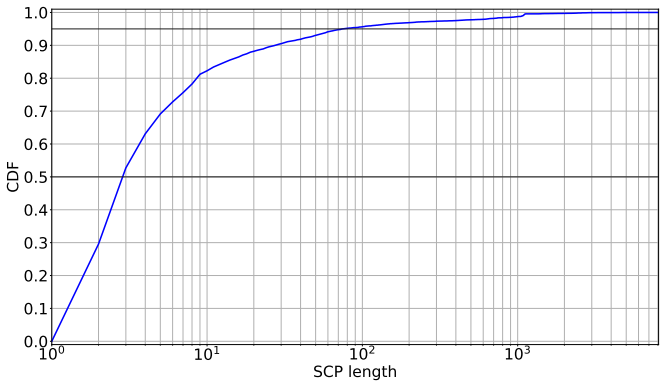
Fig. 5: Some potential scenarios of SCP shown in different colors, whereas the packet counter is indicated above each packet.

V. Experimental Results

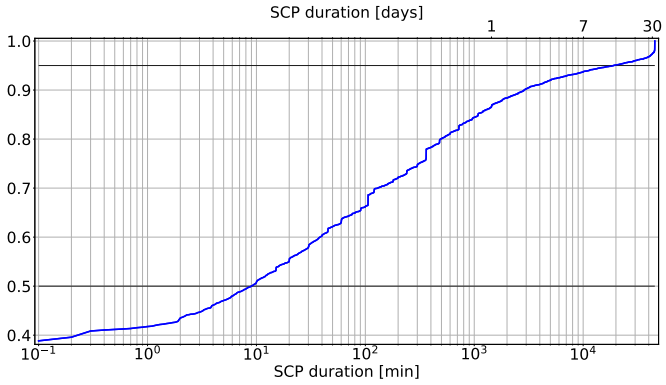
To check the feasibility of utilizing the demonstrated SCP algorithm on real data, the acquired data from the implemented packet sniffer are taken as an example. Using

Data: $\mathcal{D}_a$
Result: $\mathcal{P}_a^l$
$i = 0$
$j = 0$
$l = 0$
$\mathcal{P}_a^l \leftarrow \mathbf{m}_a^l[j] \leftarrow \mathbf{m}_a[i]$
for $i = 0$ *to* $N_a - 2$ **do**
  $FCnt_{diff}$ = $(FCnt[i+1] - FCnt[i])$
  C_1 = $(\frac{FCnt[i]+2}{FCnt[i+1]+1} \geq 0.75\ )$
  C_2 = $(0 \leq FCnt_{diff} \leq 1)$
  C_3 = $(2 \leq FCnt_{diff} \leq 21)$
  C_4 = $(MT[i] == MT[i+1])$
  C_5 = $(ADR[i] == ADR[i+1])$
  C_6 = $(SF[i] == SF[i+1])$
  C_7 = $(PS[i] == PS[i+1])$
  $conditions$ =
   $C_1 \wedge (C_2 \vee (C_3 \wedge (C_4 \wedge C_5 \wedge (C_6 \vee C_7))))$
  if $conditions == True$ **then**
    $j = j + 1$
  else
    $j = 0$
    $l = l + 1$
  end
  $\mathcal{P}_a^l \leftarrow \mathbf{m}_a^l[j] \leftarrow \mathbf{m}_a[i+1]$
end

Algorithm 1: A heuristic of SCP initial identification phase



(a) SCP length and its corresponding CDF.



(b) SCP duration and its corresponding CDF.

Fig. 6: Characteristics of the formed SCPs from the measurement.

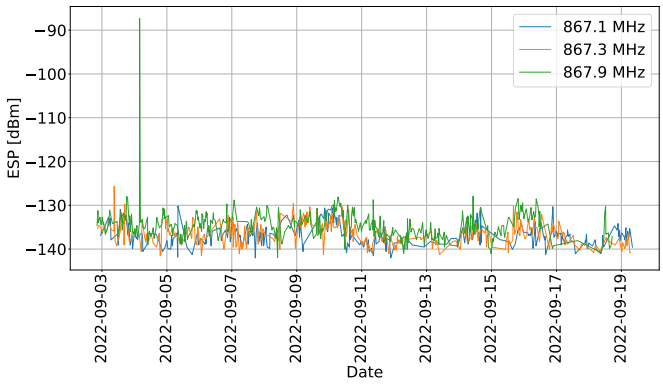
the proposed algorithm, a total number of 10779 SCPs are detected. As shown in Figure 6a, the CDF curve shows that roughly 50% of the formed SCPs are composed of less than 3 packets, while less than 5% of them have more than 100 packets. Accordingly, Figure 6b indicates that roughly 50% of the estimated SCPs have a duration of less than $10\,\mathrm{min}$, while less than 10% of them last for more than 7 days. Indeed, these percentages are subjected to the heuristic parameters chosen for the presented algorithm. Otherwise, the obtained results are primarily driven by the type of activation method (OTAA or ABP) as well as the transmission pattern of the monitored applications.

Among the presented results, 3 different SCPs with long duration ($>$ 7 days) are investigated with respect to their ESP values. As shown in Figure 7, it is observed that the ESP values are homogeneous for each different SCP $\mathcal{P}_a^l$. For example in Figure 7a, it is observed that the ESP variation is relatively small which most properly corresponds to a static end node during its SCP $\mathcal{P}_{0E15D035}^{6866}$. Moreover, its Channel State Information (CSI) is almost flat, on contrary, Figure 7b shows a clear frequency selectivity, particularly at $867.9\,\mathrm{MHz}$ with a deep fade of more than $5\,\mathrm{dB}$ depth. On the other hand, the ESP values in Figure 7c have very different fluctuation regimes which reveal a highly changing environment model, or most probably an end node moving across different locations.

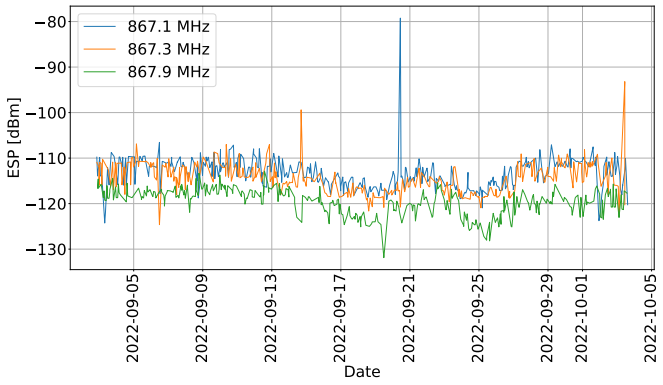
VI. Conclusion

This paper introduces a passive packet sniffer for monitoring the active LoRa devices at a given geographical location. For this purpose, a measurement campaign is carried out in the city of Rennes to collect LoRa packets using a gateway and some commodity tools. Hence, different transmission parameters are acquired while information from the packets' fields is obtained by decoding the payload header. After having the device identifiers (DevAddr and DevEUI), some vital information, such as the network operators and devices' manufacturers, is revealed and then statistically analyzed. On the other hand, the proposed SCP algorithm detects streams of successive packets transmitted from the same end device, after applying it to the acquired data. The practicality of such packet classification is demonstrated through analysis of the measured ESP values and the statistical characteristics of the estimated SCPs.

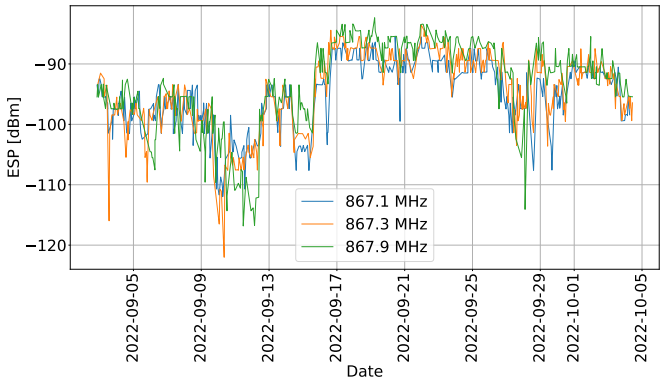
For future work, the presented packet sniffer can be used to troubleshoot and evaluate the operation of the LoRaWAN networks. Moreover, the demonstrated SCP identifier can be used to monitor all the nearby active devices and to track any targeted ones. For Research & Development purposes, extra



(a) ESP values of SCP $\mathcal{P}^{6866}_{0E15D035}$ across time.



(b) ESP values of SCP $\mathcal{P}^{7029}_{0E2EC31B}$ across time.



(c) ESP values of SCP $\mathcal{P}^{8847}_{0F47C7E6}$ across time.

Fig. 7: Some examples of formed SCP $\mathcal{P}^{l}_{a}$.

information can be estimated for each SCP, such as the packet inter-arrival time, transmission pattern, ESP values and used frequencies.

ACKNOWLEDGMENT

The authors would like to thank Jean François Legendre from Gwagenn company for borrowing from him the gateway [19].

REFERENCES

[1] A. Abdelghany, B. Uguen, C. Moy and D. Lemur, "Decentralized Adaptive Spectrum Learning in Wireless IoT Networks Based on Channel Quality Information," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19660-19669, 15 Oct.15, 2022.

[2] R. Oliveira, L. Guardalben and S. Sargento, "Long Range Communications in Urban and Rural Environments," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 810-817, 2017.

[3] Addressing & Activation | The Things Network. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/addressing/.

[4] N. Blenn, and F. Kuipers, "LoRaWAN in the wild: Measurements from the things network," *arXiv preprint*, arXiv:1706.03086, 2017.

[5] P. Spadaccino, F.G. Crinó and F. Cuomo, "LoRaWAN Behaviour Analysis through Dataset Traffic Investigation," *Sensors 2022*, vol. 22, no. 7, p. 2470.

[6] L. Bhatia, M. Breza, R. Marfievici and J.A. McCann , "Dataset: LoED: The lorawan at the edge dataset," arXiv preprint arXiv:2010.14211, 2020.

[7] K.N. Choi, H. Kolamunna, A. Uyanwatta, K. Thilakarathna, S. Seneviratne, R. Holz, M. Hassan and A.Y. Zomaya, "LoRadar: LoRa sensor network monitoring through passive packet sniffing," *ACM SIGCOMM Computer Communication Review*, pp. 10-24, 2020.

[8] Tektelic KONA Macro IoT Gateway. [Online]. Available: https://www.tektelic.com/uploads/Brochures/Kona%20Macro.pdf.

[9] ChirpStack open-source LoRaWAN Network Server. [Online]. Available: https://www.chirpstack.io/.

[10] MQTT - The Standard for IoT Messaging. [Online]. Available: https://mqtt.org/.

[11] Node-RED. [Online]. Available: https://nodered.org/.

[12] LoRaWAN packet decoder. [Online]. Available: https://github.com/anthonykirby/lora-packet.git.

[13] Measurement Data. [Online]. Available: https://gitlab.com/ahmednagy/lorawan-beaulieu-measurement-2022.git.

[14] A. Abdelghany, B. Uguen, C. Moy and D. Lemur, "On Superior Reliability of Effective Signal Power versus RSSI in LoRaWAN," *28th International Conference on Telecommunications (ICT)*, London, United Kingdom, pp. 1-5, 2021.

[15] F. Turčinović, J. Vuković, S. Božo and G. Šišul, "Analysis of LoRa Parameters in Real-World Communication," *International Symposium ELMAR*, pp. 87-90, 2020.

[16] The IEEE Standards Association. [Online]. Available: https://standards-oui.ieee.org/oui/oui.txt.

[17] NetID and DevAddr Prefix Assignments | The Things Network. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/prefix-assignments/.

[18] A. Hagberg, D. Schult and P. Swart, "Exploring network structure, dynamics, and function using NetworkX," *Proceedings of the 7th Python in Science Conference (SciPy2008)*, (Pasadena, CA, USA), pp. 11–15, Aug 2008.

[19] Gwagenn company website. [Online]. Available: http://www.gwagenn.com/en/home/.