Enabling Edge processing on LoRaWAN architecture

Stefano Milani,
Ioannis Chatzigiannakis
University of Roma "La Sapienza"
Rome, Italy
stefano.milani@uniroma1.it
ichatz@diag.uniroma1.it

Domenico Garlisi
University of Palermo
Palermo, Italy
domenico.garlisi@unipa.it

Matteo Di Fraia,
Patrizio Pisani
UNIDATA S.p.A.
Rome, Italy
m.difraia@unidata.it
p.pisani@unidata.it

ABSTRACT

LoRaWAN is a wireless technology that enables high-density deployments of IoT devices. Designed for Low Power Wide Area Networks (LPWAN), LoRaWAN employs large cells to service a potentially extremely high number of devices. The technology enforces a centralized architecture, directing all data generated by the devices to a single network server for data processing. End-to-end encryption is used to guarantee the confidentiality and security of data. In this demo, we present Edge2LoRa, a system architecture designed to incorporate edge processing in LoRaWAN without compromising security and confidentiality of data. Edge2LoRa maintains backward compatibility and addresses scalability issues arising from handling large amounts of data sourced from a diverse range of devices. The demo provides evidence on the advantages in terms of reduced latency, lower network bandwidth requirements, higher scalability, and improved security and privacy resulting from the application of the Edge processing paradigm to LoRaWAN.

KEYWORDS

Edge processing, LoRaWAN, end-to-end security

1 INTRODUCTION

In recent years, wireless technologies and mobile-generated traffic, including IoT, have rapidly expanded, becoming the largest segment of internet traffic. LoRaWAN, a widely adopted LPWAN technology, is an ideal solution for connecting various IoT devices with minimal infrastructure requirements [1]. In LoRaWAN the GateWays (GWs) have the role of simply forwarding all the traffic between the terminals, the End Devices (EDs), and the central Network Server (NS). The traffic is then forwarded to the designated Application Server (AS) that securely handles, manages and interprets the application data. The Join Server (JS) is responsible for the activation of the ED [2]. LoRaWAN supports two methods for registering and activating EDs on the network: i) Over-the-Air Activation (OTAA) and ii) Activation by Personalization (ABP), employing a secure communication protocol with encryption and authentication to ensure network security and privacy [3]. In ABP the session keys are pre-configured while in OTAA the session keys are generated during the ED activation phase.

The centralised architecture and the provided activation methods of LoRaWAN forces the processing of the frames to be carried out exclusively in the cloud. Using the LoRa GWs as simple bridges places a lot of pressure on the central NS that needs to support a massive number of ED. This presents a substantial limitation on system performance in terms of scalability and can also impact time-sensitive IoT services.

2 EDGE2LORA

In this demo, we propose Edge2LoRa a new LoRaWAN-based architecture to support edge processing that builds upon the OTAA [4]. Edge2LoRa consists of several elements: the device registry, the gateway selection algorithm, the group key establishment method and the sensor data stream processing. Each ED is serviced by one E2GW that carries out the data processing tasks on the received sensor data streams. Edge2LoRa operates over the conventional LoRaWAN architecture, ensuring backward compatibility. Access to the data within the frames is facilitated by establishing a group key among the cloud (AS), edge (E2GW), and end device (E2ED). Two shared session encryption keys are created between the E2ED, E2GW and AS: i) the Edge Session Encryption Key and the Edge Session Integrity Key. The former is used to enable secure encryption and decryption of the frame payload. The latter is used to check the integrity of the edge-specific frames. In this context, the proposed approach employs elliptic curve cryptography for generating cryptographic keys [5].

3 HARDWARE SET-UP AND SOFTWARE MODULES

Fig.1 depicts the hardware architecture used for the demo, this includes 2 real *LoRaWAN GW*, implemented by using a Raspberry Pi 3 piloting the GW transceiver composed by a SEMTECH SX1301 chip, receive over-the-air the signal produced through 5 HTCC-AB01 "CubeCell" series terminals, integrated the PSoC 4000 series MCU and SEMTECH SX1262 chip. Additionally, for each terminal, we connect a
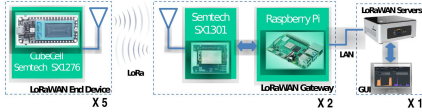
well as the AS, which also hosts the demo Graphical User Interface (GUI). Finally, Fig.2 shows the selected protocols to interface the different modules.

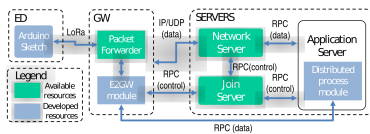**Figure 1: Demo hardware architecture.**

Figure 2: Edge2LoRa flow diagram.

temperature/humidity/pressure sensor (BME280) to generate stream data. NS, AS and JS are deployed by an Intel NUC 5i5RYH Mini PC (8 GB of RAM). Additional monitor, keyboards and mouses (not shown in the figure), will be utilized for visualizing and controlling the demo dashboard.

Fig.2 summarizes the software architecture of the demo. In the figure, according to the legend, two types of software modules are considered, the module already available as open source software and the module implemented to build the Edge2LoRa architecture. It is important to highlight that the resulting system maintains backward compatibility, allowing seamless interoperability between legacy and new components, as demonstrated in the demo. Three main blocks can be recognized in the figure: i) the ED including the Arduino Sketch module; ii) the GW including two modules, the Packet Forwarder (the legacy SEMTECH GW module) and the E2GW module, iii) the SERVERS, including the NS and JS (both based on TheThingsStack), as

4 DEMONSTRATION DESCRIPTION

The objective of the demonstration is to provide insights on the Edge2LoRa implementation in a real LoRaWAN network and show the performance gains under different scenarios, i.e. under different EDs configurations and data aggregation. The 2 GWs are configured as legacy GW the first and as E2GW the other. Legacy GW follows the classical data flow, while the E2GW enables multiple aggregation functions and direct connection to the AS (exclusively for the traffic generated from the terminals configured as E2ED). The demo GUI enables two different control sections, the first has been designed to control the activity of the terminals and each terminal can be tuned as legacy or E2ED. The terminal source rate and the message payload can be configured by the onboard button console. The second GUI section receives the control of the applications in terms of aggregation function selection (including mean, sum, max and min of the sensor data) and window time. Moreover, the dashboard also tunes the available bandwidth of the link connections between GWs and AS, Indeed, the benefit of the aggregation is more evident when the link is slow. Fig.2 also illustrates how the different elements' architectures are connected. For clarity, we describe only the uplink traffic. Data generated from the EDs follows the classical flow through the legacy GW, which forwards the data to the NS. Conversely, in the E2GW, only frames from E2ED activate the E2GW module, where data stream operators apply transformations that are sent directly to the AS. The system uses many-to-one transformations, potentially aggregating data received from multiple frames. According to the radio coverage of the GWs and to the frames collision occurrence, only a subset of frames will be duplicated in the system. For this reason, Edge2LoRa relies on a duplicate detection filter (DDF) for identifying whether a given frame has previously appeared in a stream of data. The DDF is maintained by the AS and identifies with no errors duplicate frames in constant time [6]. The AS upon receiving a frame from the NS will assign a timeout before processing it. The timeout is set in a way such that it will allow the E2GW to complete the processing of the operator.

Finally, a visualization section is present in the dashboard to monitor system results, the number of frames received from the legacy path as well as from the Edge2LoRa path and network statistics for: i) *latency*: we show that by processing data closer to the source we reduce the latency associated with sending data to the cloud; ii) *network traffic*: network traffic is significantly reduced by aggregating data at the Edge; iii) *scalability*: in terms of generated data, here distributed computing resources can be easily scaled up or

	Legacy	Edge2LoRa	GAIN
Latency	$955 \pm 4.6\ ms$	$745 \pm 7.3\ ms$	$210\ ms$
Data	$120\ kBytes$	$24\ kBytes$	$96\ kBytes$

Table 1: Comparison results between legacy and Edge2LoRa in terms of latency and data traffic.

down based on the demand, without requiring additional infrastructure; iv) *security and privacy capabilities*: configured message by EDs control can be visualized at the GW point, ciphered and in clear format. Data are processed at the Edge thus reducing the risk of large-scale data breaches and privacy violations. In Tab.1, we present performance results in terms of latency and network traffic for a scenario with 2 EDs, a legacy device, and an E2ED. We measure the average time difference between when the frames leave the Packet Forwarder module and when they reach the AS. The table compares the classical approach (Legacy column) to Edge2LoRa. Notably, Edge2LoRa enhances system performance by reducing latency from $955 \pm 4.6ms$ to $745 \pm 7.3ms$ (95% Confidence Intervals) and data traffic from $120KB$ to

$24KB$. These results were obtained during a period of activity where the EDs sent 100 frames, and the chosen aggregation function considered windows of 5 frames.

REFERENCES

[1] Antonino Pagano, Daniele Croce, Ilenia Tinnirello, and Gianpaolo Vitale. A survey on lora for smart agriculture: Current trends and future perspectives. *IEEE Internet of Things Journal*, 10(4):3664–3679, 2023.

[2] LoRa Alliance. Lorawan 1.1 specification. *technical specification*, 2017.

[3] Ismail Butun, Nuno Pereira, and Mikael Gidlund. Analysis of lorawan v1.1 security. pages 1–6. ACM Press, 2018.

[4] Stefano Milani and Ioannis Chatzigiannakis. Design, analysis, and experimental evaluation of a new secure rejoin mechanism for lorawan using elliptic-curve cryptography. *Journal of Sensor and Actuator Networks*, 10:36, 6 2021.

[5] K. Lauter. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless Communications*, 11:62–67, 2 2004.

[6] Rémi Géraud-Stewart, Marius Lombard-Platet, and David Naccache. Approaching optimal duplicate detection in a sliding window. In *Computing and Combinatorics: 26th International Conference, COCOON 2020, Atlanta, GA, USA, August 29–31, 2020, Proceedings 26*, pages 64–84. Springer, 2020.