Proximeter CERN's detecting device for personnel

Christoph Merscher[1,*], *Rodrigo* Sierra[1,**], *Alessandro* Zimmaro[1,***] *Marco* Giordano[1,****], and *Salvatore* Danzeca[1,†]

[1]CERN

Abstract. TheSARS COV 2 virus, the cause of the better known COVID-19 disease, has greatly altered our personal and professional lives. Many people are now expected to work from home but this is not always possible and, in such cases, it is the responsibility of the employer to implement protective measures. One simple such measure is to require that people maintain a distance of 2 metres but this places responsibility on employees and leads to two problems. Firstly, the likelihood that safety distances are not maintained and secondly that someone who becomes infected does not remember with whom they may have been in contact. To address both problems, CERN has developed the "proximeter", a device that, when worn by employees, detects when they are in close proximity to others. Information about any such close contacts is sent securely over a Low Power Wide Area Network (LPWAN) and stored in a manner that respects confidentiality and privacy requirements. In the event that an employee becomes infected with COVID-19 CERN can thus identify all the possible contacts and so prevent the spread of the virus. We describe here the details of the proximeter device, the LPWAN infrastructure deployed at CERN, the communication mechanisms and the protocols used to respect the confidentiality of personal data.

1 Introduction

The spread of the COVID-19 virus poses major challenges for all branches of the economy. However, everyone agrees on one point: the protection of employees has top priority. For this reason, several measures have been adopted at CERN to help protect employees as much as possible. One of these measures is the rule that employees should always keep a distance of 2 meters from each other. Unfortunately, this is not always feasible in practice. There may be situations where keeping a distance is not possible. In addition, it must also be expected that these distances, intentionally or unintentionally, are not always maintained. In the event that an employee is tested positive for COVID-19, it is imperative that all persons who have been in contact with the infected person are informed. This is to reduce the risk of infection at CERN. It may be difficult for a person who has been infected with the virus to remember all contacts of the last days. Furthermore, it is quite possible that a person is so strongly affected

[*] e-mail: christoph.merscher@cern.ch
[**] e-mail: rodrigo.sierra@cern.ch
[***] e-mail: alessandro.zimmaro@cern.ch
[****] e-mail: marco.giordano@cern.ch
[†] e-mail: salvatore.danzeca@cern.ch

by the virus that a concrete questioning is no longer possible. For this very reason, CERN has launched the proximeter project. The idea of the project is quite simple. A small device, no bigger than a credit card but 2cm thick, has to be worn by every CERN employee at all times. If two employees come closer than 2 meters for more than 30 seconds, the proximeter registers this and sends the information to a corresponding database. However, no geo data are stored but only ID's which can be assigned only by the medical service to an employee. The backbone of the system is not WLAN as usual but a new IoT network, LoRaWAN. This has a longer range and makes it possible to have end devices in operation longer, since this protocol is more resource-efficient. Various groups at CERN worked closely together in the planning and development of the proximeter to quickly present a solution. In this paper we present the network installed at CERN and used for the proximeter project as well as the proximeter terminal itself.

2 LoRaWAN

Low Power Wide Area Network (LoRaWAN) is only one of several LPWAN protocols. It enables wireless, battery-powered devices to connect to the Internet. LoRaWAN[1] has been chosen as default IoT proctol for CERN over other protocols because of its maturity, good radio and capacity characteristics, availability of devices and chips, back-end features and the possibility of integration with other services. It is precisely these advantages of LoRaWAN that ultimately benefit the proximeter project. Although the protocol was not explicitly selected for the proximeters, it is optimally equipped for it. A more detailed explanation of why CERN chose LoRaWAN can be found in Rodrigo Sierra's *Readying CERN for connected device era*[1].

An interesting question is why CERN decided to realize the proximeter project with an Internet of Things (IoT) network instead of focusing on old proven networks like Wireless Local Area Network (WLAN) ? Three factors tipped the scale; service availability, power consumption and confidentiality. LoRaWAN, as stated by its name, offers extremely long-range data links which allows to exchange data all around the campus (around 60 km^2) with minimal deployment. Wi-Fi is available in all offices but does not cover all technical buildings nor the outdoors areas. Secondly, because the proximeters are battery-powered and are in sleep mode except when they are sending data, LoRa has a reduction in power consumption by a factor of 10 compared to Wi-Fi. Another reason to choose LoRaWAN was anonymity. The LoRaWAN parameters are only known by the LoRaWAN network administrators - not even the owner of the device knows them – and all communications are encrypted from the device up to the application.

The LoRaWAN protocol makes it possible to achieve a coverage of several kilometers, but this also brings some limitations. One of these limitations relates to the possible size of data packets, which is severely restricted. How large individual data packets can be depends on the so-called Spreading Factor (SF). The larger the SF, the further the device can reach the network but the longer individual packets remain in the air -the lower the data rate-. Table 1 shows how large data packets can be depending on the spreading factor.

It should also be noted that a header, which is attached to each message as a prefix, contains 13 bytes. To ensure that all the required information can still be transmitted, it is advisable to encode them. Furthermore, it should be determined beforehand which SF are permissible. As the network capacity depends on the spreading factors allowed -the higher

[1] https://lora-alliance.org/

the SF, the lower the number of packets the network can process- the network operators need to balance the maximum allowed packet size, spreading factor and packets per device. The counterpart to take into account is coverage; the lower the maximum allowed SF, the lower the coverage of the network. For the proximeter project SFs equal or higher to 10 were chosen as a compromise between indoors coverage, data rate and duty cycle constraints, that will be explained later in this section. This means that the data packets must not exceed 51 bytes.

Another problem that arises when using LoRaWAN is that it is an open radio network in the unlicensed Industrial, Scientific and Medical(ISM) band. Being an open radio network, the network will receive the LoRaWAN communications from anyone using the same technology in the area. To ensure that the network server only evaluates messages sent by the user's own end devices, they must be registered in the application server. There are two variants for the registration and the subsequent join request:

1. **Over-the-Air Activation (OTAA):** is the preferred and most secure way to connect with a network server. Devices perform a join-procedure with the network, during which a dynamic Device Address is assigned and security keys are negotiated with the device.
2. **Activation by Personalization (ABP):** In some cases it can not be avoided to hardcode the Device Address as well as the security keys in the device. This means activating a device by personalization. This strategy might seem simpler, because it skips the join procedure, but it has some downsides related to security.

At CERN, the OTAA principle is used. The security keys which are sent with the join request are the following:

- **Device EUI:** is a 64-bit globally-unique Extended Unique Identifier (EUI-64) assigned by the manufacturer, or the owner, of the end-device.[3]
- **Application key:** is a 128-bit key which is only known by the end device and the application itself. It is used to derive session keys during the activation procedure.

Only if these two keys match the values stored in the application server, the device can connect to the network.

Using an unlicensed ISM band, LoRaWAN shares the spectrum (and the possible interference) with any other radio protocol using the same frequencies. The European Telecommunications Standards Institute (ETSI) imposes a duty cycle in this band for all transmitters. All LoRa end devices are subject to this duty cycle, which means that a device may only send data for 1% of the time or, in other words, 36 seconds per hour. As the restriction applies to the transmitter, the end devices themselves are responsible for ensuring that this regulation is complied with as the network cannot control it. This regulation applies not only to end devices but also to gateways, although their duty cycle is a little higher as

Spreading Factor	Max packet size (in bytes)
12,11,10	51
9	115
8,7	222

Table 1. Max packet size per SF for LoRaWAN networks[2]

they can use different sub-band. It is therefore the responsibility of the network to monitor the duty cycle compliance of the gateways.

In order to use the LoRaWAN network at CERN, it was necessary to provide an appropriate architecture, which is shown in Figure 1. The initial back-end selection [1] was re-evaluated following the new requirements introduced by the proximeter project. ChirpStack[2] was finally chosen. A more detailed explanation of the evaluation process will not be given here, as it would exceed the scope of this paper. The various cornerstones of the architecture shown in figure 1 will nevertheless be briefly discussed:

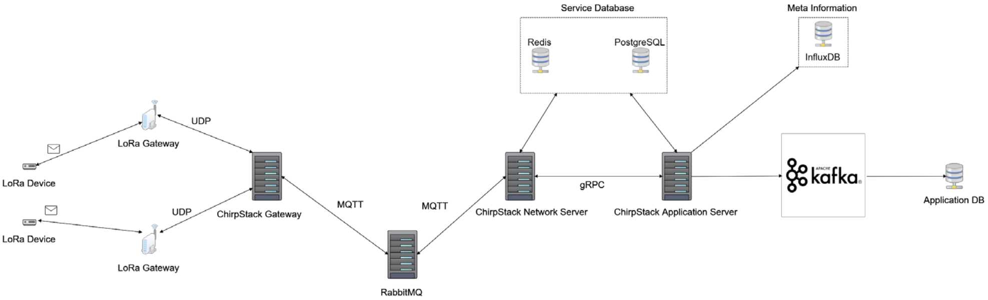

Figure 1. LoRaWAN at CERN

- **LoRa Device:** This can be any device that supports the LoRa standard. Currently, only class A devices are used at CERN. Class A in this context means that communication is always initiated by the end-device and is fully asynchronous. Each uplink transmission can be sent at any time and is followed by two short downlink windows, giving the opportunity for bi-directional communication, or network control commands if needed.[**?**] The actual data are transmitted to the gateways using radio waves.
- **LoRa Gateway:** With the help of these gateways it is possible to receive data from LoRa end devices and to transmit them to a network server using established network technologies, in this case User Datagram Protocol (UDP). At CERN, Lorix^{One}[3] gateways are used, which offer an open and customized Linux system that allows additional security measures (such as restricting access via a firewall) to be enforced.
- **ChirpsStack Gateway Bridge:** ChirpStack Gateway Bridge is a service which converts LoRa Packet Forwarder protocols into a ChirpStack Network Server common data-format (JSON and Protobuf). This component is part of the ChirpStack open-source LoRaWAN® Network Server stack.[4]
- **ChirpStack Network Server:** The network server is responsible for de-duplication of messages which can occur when a LoRa device sends a message to more than one gateway. Furthermore, the network server handles authentication, mac-layer and mac-commands, communicates with the application server and schedules downlinks (messages sent from the network to the end devices).
- **ChirpStack Application Server:** This server is responsible for the inventory, which means that it manages the devices that can connect to the server. In addition, it also manages the join requests and the encryption of the application payloads.

[2] https://www.chirpstack.io
[3] https://www.lorixone.io/

- **Service Database:** These databases contain data which are indispensable for the operation of the server. Volatile information, such as the device address, which is regenerated each time a join occurs, is written to the Redis database. Permanent data sets, such as the device EUI or the application key, are persisted in the PostgreSQL database. In summary, these databases contain all necessary information about all registered LoRa end devices as well as those of the gateways.
- **Meta Information:** This database is used to store meta information such as the SF or which gateway forwarded the message. The goal is to continuously improve the quality of the network by collecting meta information. This database is managed purely for support reasons and is not necessary for the actual operation of the network.
- **Application DB:** Finally, the data sent by LoRa devices and received by the application server is persisted in an application database. From there, different applications, which have to be developed separately, can access the data and use it according to the specific use case.

One point that has not yet been discussed but also plays an important role concerns the downlink. How can data packets be sent from the network server to the end devices and is this supported by the protocol at all? In principle, the LoRaWAN protocol supports sending messages to LoRa end devices. Due to the lack of frequency band separation between uplink and downlink communication, LoRa gateways operate in half-duplex mode, hence they cannot receive and transmit simultaneously. When a frame is transmitted, all ongoing receptions are aborted and no uplink frame can be received for the entire duration of the downlink frame. The presence of downlink traffic in a large network can therefore lead to loss of data packets.

Since it is important for the proximeter project to ensure that the data packet has actually arrived, an ACK mechanism is used. This means that a message notifies the end device as soon as the network server has successfully received the data packet. At this point it is important to note that in the LoRaWAN area it is generally not recommended to acknowledge the receipt of data packets, as this traffic places an additional load on the network. As discussed earlier in this section, only class A devices are used for the proximeter project and this means that traffic is always initialized by the device itself. Accordingly, the question arises as to how the downlink concept is technically implemented? Downlink messages from the network server are queued until the next time an uplink message is received from the end device and a receive window (Rx) is opened. This design is specifically geared toward applications that require downlink communication in response to an uplink, or that can schedule downlinks ahead of time with fairly loose latency requirements. Following an uplink, a Class A end device opens a short receive window (Rx1) and, if no downlink is received during that period, it opens a second receive window (Rx2). The start time of Rx1 begins after a fixed amount of time following the end of the uplink transmission. Typically, this delay is one second, however this duration is configurable. Rx2 typically begins two seconds after the end of the uplink transmission, though this duration is also configurable. [5]

Figure 2 help to better understand the situation. In this picture it can be seen that both Rx windows were opened but no message was received from the end device. If the end device had received a message in Rx1, Rx1 itself would have been larger and Rx2 would not have been opened at all. If the end device received the message not in Rx1 but in Rx2, the Rx2 section would also be larger.

To ensure that the network at CERN is up to the challenges, the maximum amount of uplink/downlink messages that the network should support according to the estimated user

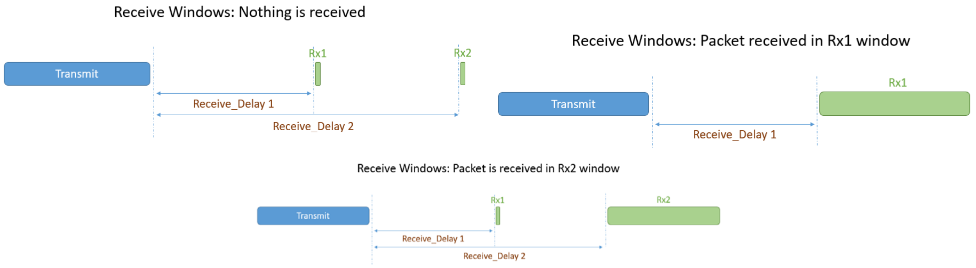
Receive Windows: Nothing is received
Transmit
Rx1
Rx2
Receive_Delay 1
Receive_Delay 2
Receive Windows: Packet received in Rx1 window
Transmit
Rx1
Receive_Delay 1
Receive Windows: Packet is received in Rx2 window
Transmit
Rx1
Rx2
Receive_Delay 1
Receive_Delay 2

Figure 2. Receive Window

behavior was calculated, and both the devices and the network were designed to cope with it. In particular, 51 gateways were installed on the site (including tunnels and LHC points). Multiple gateways, inside and outside the building, were installed in hot spots such as restaurants.

3 Proximeter

In this section we describe the end node device from a hardware and a software perspective. The components used, as well as important firmware features, are pointed out and explained thoroughly.

3.1 Hardware

The Proximiter device is a twofold design: it is composed of a mainboard and a sensor-board. The sensor-board hosts a Qorvo $DWM1001C$ Ultra Wide Band (UWB) sensor, featuring a microstrip antenna tuned on channel 5, with a frequency of 6.5 GHz and a band of 500MHz. The sensor-board communicates with the mainboard via an SPI bus, that is shared with the LoRa transceiver (RF96 from HopeRF Electronics), augmented with a ceramic antenna ($ANT-868-USP$ from Linx Technologies), a 128Mb flash memory ($IS25LP128-JBLE$ from ISSI) and an external Real Time Clock ($DS1347$ from Maxim Integrated), which are all hosted on the mainboard. The core of the system is a $SAMD21G18A$ from Microchip that embeds an ADC, timers, serial communications (UART, SPI) and a USB interface. The MCU is based on a $Cortex-M0+$ architecture and is running at 48MHz clocked via a DFLL (Digital Frequency Locked Loop). The board is powered by a rechargeable Li-Ion battery ($NCR18650B$ from Samsung) rated at 3350mAh. The recharge is managed by a $BQ24230RGTR$, from Texas Instrument. It is configured to charge at 500mA with an end voltage of 4.2V. The power conversion is done by an LDO, $ADP124$ from Analog Devices, and the power is delivered to the board by a smart push button controller, $STM6601$ from ST Semiconductor. Concluding, the device also endows a red LED, which is used to assess the charging state, an RBG status LED, and a vibrator motor, which warns the user when he/she is within the specified range, for more than 30 seconds.

3.2 Software

When powered on, the MCU initializes the peripherals and the flash memory. The latter is divided in two partitions, one sector (4096 Bytes) is used for configuration while the rest,

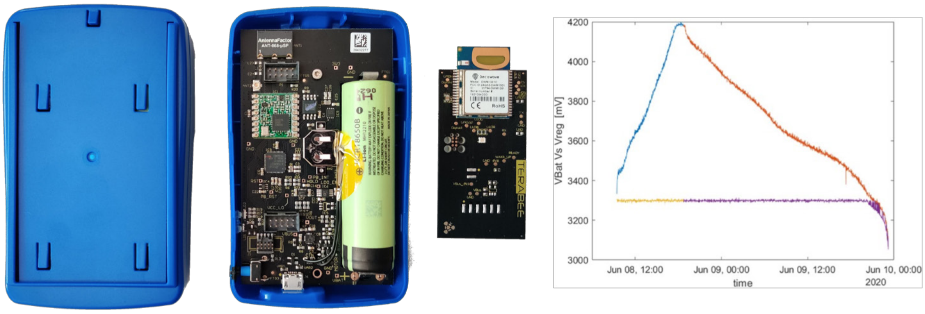

Figure 3. Proximiter enclosure and internal components. Charge and discharge voltage profile.

implemented as a circular buffer, is used for data storing. After the initialization, data contained in both partitions are loaded. Once the initialization phase is ended, old encounters, if present, are sent via LoRaWAN in the first time slot available. Then, the MCU enters in a continuous loop where it starts the join procedure and to record encounters. During the join, the green LED is turned steadily on, while, when the device is connected, the green LED is blinking. When the sensor has a new distance measure available, it rises a GPIO to report the availability of the data to the MCU. The latter sends an SPI ”Get Command” and wait for an ack from the sensor via a GPIO pin. The MCU reads the distance measured and its relative TAGID that allows to identify the encountered device. The encounter data consists of 3 information: the TAGID, the number of 30 seconds windows the encounter lasted, and information about the start time (day, hour and minute) of the encounter encoded in 2 bytes. At the end of the encounter, the recorded information are saved in the external flash, and will be read to compose the LoRa packet. The LoRa packet can contain up to 7 records, and every 15 minutes a LoRaWAN transmission is scheduled if there are ended encounters not sent. We exploit both the uplink and downlink of the LoRaWAN protocol, and we reliably send LoRaWAN packets using the “confirmed uplink”. In case of missed ACK, we resend the same packet in the next available window. The access to the device is encrypted via AES-128, and so is the bootloader, which can be activated programmatically. Firmware Update Over The Air (FUOTA) will be addressed as future work on the device.

4 Anonymity

In order to guarantee the anonymity of the end users, the data sent by the proximeters to the network is processed according to a certain scheme. The data is transmitted, encrypted, from end devices to LoRa gateways, which, in turn, transmit the data to the network server, which forwards it via grpc to the application server, which is also encrypted. From there the data is forwarded directly to Kafka from where it is persisted in a database that is not visible to the network operator. Although meta information is stored in a separate database for the purpose of improvement, it does not contain any information that would allow conclusions to be drawn about the actual user. In addition, the database that contains the actual information does not store the Device EUI, which allows a unique identification of the device, but only the name of the device as it is stored in the application server. Thus, information about end devices is stored in different databases, each of which is operated by different groups. No single group has a complete overview of the data and can therefore not trace which user which data originates from. Only the Medical Service has access to this data and, in the event of a positive Corona test of an employee, can track which colleagues the infected person came into contact with and contact them accordingly.

5 Conclusion

The consequences of the COVID-19 pandemic pose major challenges for all of us. Regular work as we are all used to is not possible for reasons of general security, which is why many companies rely on tele working strategy. Unfortunately, this is not always possible, as certain professions cannot work remotely, but it is important to protect them as well as possible. For this very reason, various rules have been formulated at CERN, one of which states that employees should always keep a distance of 2 meters from each other. Since this is not always observed for various reasons, the proximeter project was launched. With the help of this device it is possible to trace with which persons an infected person came into contact without storing geographically specific data. This involves the use of a new network infrastructure, LoRaWAN. This technology makes it possible to cover the entire CERN campus without much effort, even where there is no or only poor WLAN reception. The network itself was not built explicitly for the proximeter project, a test phase was already underway, but the project has rapidly pushed the expansion of the network forward. It should be mentioned that the project uses the LoRaWAN protocol in a hacky variant. This is especially due to the fact that all messages sent by proximeters are acknowledged. Although the protocol supports the confirmation of messages, it is strongly discouraged to use it. LoRa is not suitable for critical applications because it is not possible to ensure that messages reach the recipient. To date, no bottlenecks have been detected on the network, although 25000 downlinks per day are now being sent over the network. Nevertheless, it must be emphasized that currently most of CERN's employees do not work on site but from home. It remains to be seen how the network will behave when all employees are back on site. In summary, it can be said that the proximeter is an effective and anonymous way to track contacts without compromising the privacy of the employees. In the event of an infection, Medical Service staff can quickly, easily and completely see which employees need to be contacted and warned. This makes it possible to keep the spread of the virus as small as possible, at least at CERN.

References

[1] H.O. Rodrigo Sierra, *Readying CERN for connected device era, journal = "edp sciences", volume = "245, 2020", number = "07015", month = nov, year = "2020", pages = "6", doi = "10.1051/202024507015", url = "https://doi.org/10.1051/epjconf/202024507015",*

[2] Lora-alliance, *About lorawan*, https://lora-alliance.org/about-lorawan/, accessed: 2021-03-11

[3] *Device eui*, https://lora-developers.semtech.com/library/tech-papers-and-guides/the-book/deveui/, accessed: 2021-02-26

[4] *Chirpstack gateway bridge*, https://www.chirpstack.io/gateway-bridge/, accessed: 2021-02-26

[5] *An in-depth look at lorawan class a devices*, https://lora-developers.semtech.com/library/tech-papers-and-guides/lorawan-class-a-devices/, accessed: 2021-02-26