LoRaWAN — Everything You Need to Know About The Global IoT Standard

What you should know before starting any LoRaWAN project.

Armando Rodrigues

Armando Rodrigues

Follow

6 min read

·

Oct 31, 2024

Listen

Share

More

Press enter or click to view image in full size

Blue and black logo, spelling LoRaWAN.

Source.

Trying to start a DIY IoT project but don’t know where to start? Or maybe just starting to learn about IoT technologies but getting overwhelmed with all the different protocols and standards? Don’t worry, I’ve got you covered! In this article I cover one of the most popular IoT standards among hobbyist and professionals: LoRaWAN, a standard built on top of LoRa. Here you’ll find some of the most important concepts on this topic. I did my best to present them in an easy-to-digest manner.

LoRaWAN Standard

LoRaWAN is a LPWAN (Low-Power Wide Area Network) standard maintained by the LoRa Alliance, an open, non-profit association with the mission of promoting the standard, which the organization claims to be the leading IoT LPWAN specification.

The LoRaWAN specification defines 3 things: the protocol stack, the regional parameters and the network architecture. The protocol stack comprises the LoRaPHY physical layer protocol and the LoRaWAN network protocol, which actually covers both the link and network layers. The network parameters specify important variables. Namely, channel frequency, channel bandwidth and the transmission windows for LoRaWAN communications all over the world. The network architecture standardizes the overall network structure, the function and naming of each device type and the relations and interactions between network nodes.

Network Architecture

Press enter or click to view image in full size

Diagram displaying network servers in the center gateways on the left and application servers on the right.

LoRaWAN network in a star topology.

LoRaWAN networks use a star topology. These networks consist of end-nodes/end-devices, gateways, network servers and application servers.

The working of gateways is explained in The Things Network (TTN) website:

Each gateway is registered (using configuration settings) to a LoRaWAN network server. A gateway receives LoRa messages from end devices and simply forwards them to the LoRaWAN network server. Gateways are connected to the Network Server using a backhaul like Cellular (3G/4G/5G), WiFi, Ethernet, fiber-optic or 2.4 GHz radio links.

Network servers manage the whole network. They filter duplicate messages received from several gateways, route uplink application payloads to the correct application server, provide acknowledgments of messages, send Adaptive Data Rate (ADR) messages, handle join requests and execute many more security- and network-related functions.

Application servers process application-layer payloads and generate downlink application messages (downlink Medium Access Control (MAC) commands, which are network and end-node control messages are generated by network servers).

End-nodes are usually made up of microcontrollers connected to sensors, actuators, or both. These devices are usually battery-powered, have LoRa modulators and implement the LoRaWAN protocol stack in firmware. Every end-device must be registered with a network before sending and receiving messages. The process of joining a network is called “end-device activation”. An end-node can be permanently tied to a pre-selected network using Activation By Personalization (ABP) or it can search and request to join a network using over-the-air activation (OTAA).

Devices that are part of a network with location-aware gateways can have GPS-free geolocation capabilities using a trilateration technique based on timestamps sent from the network.

LoRaWAN Network Protocol

Press enter or click to view image in full size

Diagram with a representation of LoRaPHY radio packets.

Representation of LoRaPHY radio packets, LoRaWAN link layer frames and network layer packets. The size of the physical radio packets is not specified because it is measured in symbols (not bytes) and it is highly variable as preamble and header length depends on the hardware (not on the LoRaWAN standard).

The LoRaWAN Network Protocol is the link and network layer protocol of the LoRaWAN protocol stack. Link layer frames are the payload of the physical layer LoRaPHY radio packets described in Subsection 2.1.2. Some frame types are used only during the join process of new end-nodes. However, the main type of link layer message, the one that is shown in Figure 2.5, contains three fields: a header, a payload and a message integrity code (MIC). Link layer frame headers include some information about the message type. Message types include confirmed and unconfirmed data messages, join and rejoin requests, MAC command and even proprietary message types, used to implement non-standard types. The payload contains the network layer packets. The MIC serves a different purpose than the CRC fields in the physical-layer protocol. Whilst CRC fields are used for error detection and correction, MIC fields are used for security purposes, allowing the receiver to determine if the message has been changed.

The link layer payload can be considered the network layer packet of the LoRaWAN specification. Still, the official documentation refers to these network layer packets as “MACPayload” frames, since the ISO/IEC 7498–1 model definitions of link and network layers do not strictly apply to LoRaWAN networks. These frames contain a frame header, an optional port field and an optional payload. The header contains the end-device device address, a frame counter, an adaptive data rate field and several other frame options fields. The port field allows for protocol testing and for future standardized application extensions. The payload contains the encrypted application-layer data.

Device Classes

Press enter or click to view image in full size

Diagram with a representation of LoRaWAN classes’ transmission windows

LoRaWAN classes’ transmission windows.

The LoRaWAN standard supports three device classes: class A (from “All End-Devices”), class B (from “Beacon”) and class C (from “Continuously Listening”). Device classes are backwards compatible, meaning that class C devices are class A and class B compatible and class B devices are class A compatible (but not class C compatible). The main difference between classes has to do with reception and transmission windows.

Class A focuses on obtaining extremely low power consumption at the cost of device availability. For devices from this class, the default state is a low-power one, such as a light sleep or even complete device hibernation. Essentially, class A end-nodes only wake for transmission, having two reception windows available after the end of transmission. The time interval between the transmission window and the reception windows is specified in the regional parameters’ specification of the LoRaWAN standard. Even though two reception windows are open after transmission, only one of them can actually be used for downlink messages, meaning that if a message is received in the first reception window the second one will not be available. Class A devices are very rarely available to receive downlinked commands from the network. Thus, most class A end-nodes are unidirectional or telemetry-oriented devices that broadcast data without accepting or accepting very few, commands. These include meters such as water, temperature, noise, current, voltage and light sensors or monitoring devices like parking, Time of Flight and motion sensors. Class A devices are commonly included in time-insensitive and loss-tolerant applications.

Class C is the complete opposite of class A. It focuses on device availability and neglects energy saving. Unlike most class A end nodes, devices from this class are usually connected to mains power and are not battery-powered. They are continuously available for reception, except during transmission. Class C end-nodes usually need to receive large amounts of data or several commands from downstream messages and are implemented in more time and response-sensitive applications.

Class B devices are an intermediate class that aims to achieve low power consumption while still offering frequent, but intermittent, reception windows, called ping slots. Devices from this class need to search and receive network beacons that allow synchronization with the network and avoid drifts between the device’s internal clock and network timing. A device might be unable to receive beacons due to being out of a gateway’s range, the network being down or interference. In such an event, class B devices gradually widen their reception windows to accommodate a possible drift of the device’s internal clock and to try to recover the network beacon. Class B devices ought to operate “beaconless” for at least two hours before downgrading to class A operation, resetting or turning off.

The Bigger Picture — An Amazing Ecosystem for Hobbyists!

The LoRaWAN standard is part of a huge IoT ecosystem regulated by the LoRa Alliance. Companies and projects such as Meshtastic, The Things Network and many others provide everything from open-source and public to community-managed and fully private LoRaWAN networks. There are literally thousands of LoRa-compatible devices available on the market and the open-source nature of the standard makes it suitable for hobbyists. I myself have built my own LoRaWAN end nodes from the ground up and hope to document that journey in future Medium stories. Stay tuned for that!

This article is based on my master’s dissertation, where I explored the development of compact and efficient LoRaWAN end nodes with a focus on antenna miniaturization and power optimization for IoT applications.

All the diagrams in this article were created by the author.

IoT

Lorawan

Lora