lpwan
Internet-Draft
Intended status: Informational
Expires: May 1, 2017

S. Farrell
Trinity College Dublin
A. Yegin
Actility

October 28, 2016

LoRaWAN Overview
draft-farrell-lpwan-lora-overview-01

Abstract

Low Power Wide Area Networks (LPWAN) are wireless technologies covering different Internet of Things (IoT) applications. The common characteristics for LPWANs are large coverage, low bandwidth, small packet and application layer data sizes and long battery life operation. One of these technologies is LoRaWAN developed by the LoRa Alliance. LoRaWAN targets key requirements of the Internet of things such as secure bi-directional communication, mobility and localization services. This memo is an informational overview of LoRaWAN and gives the principal characteristics of this technology in order to help with the IETF work for providing IPv6 connectivity over LoRaWAN along with other LPWANs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 1, 2017.

Table of Contents

1. Introduction

LoRaWAN is a wireless technology for long-range low-power low-data-rate applications developed by the LoRa Alliance, a membership consortium. <https://www.lora-alliance.org/> LoRaWAN networks are typically organized in a star-of-stars topology in which gateways relay messages between end-devices and a central "network server" in the backend. Gateways are connected to the network server via IP links while end-devices use single-hop LoRaWAN communication that can be received at one or more gateways. All communication is generally bi-directional, although uplink communication from end-devices to the network server are favoured in terms of overall bandwidth availability.

In LoRaWAN networks, end-device transmissions may be received at multiple gateways, so during nominal operation a network server may see multiple instances of the same uplink message from an end-device.

To maximize both battery life of end-devices and overall network capacity, the LoRaWAN network infrastructure manages the data rate and RF output power for each end-device individually by means of an adaptive data rate (ADR) scheme. End-devices may transmit on any channel allowed by local regulation at any time, using any of the currently available data rates.

This memo provides an overview of the LoRaWAN technology for the Internet community, but the definitive specification [LoRaSpec] is that produced by the LoRa Alliance. This draft is based on version 1.0.2 of the LoRa specification. (Note that version 1.0.2 is expected to be published in a few weeks. We will update this draft when that has happened. For now, version 1.0 is available at [LoRaSpec1.0])

2. Terminology

This section introduces some LoRaWAN terms. Figure 1 shows the entities involved in a LoRaWAN network.

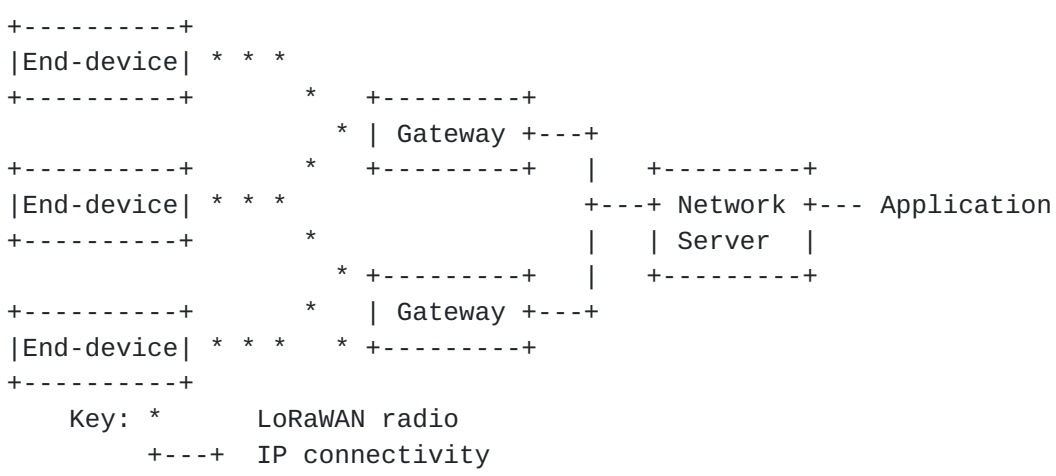

Figure 1: LoRaWAN architecture

- End-device: a LoRa client device, sometimes called a mote. Communicates with gateways.

- Gateway: a radio on the infrastructure-side, sometimes called a concentrator or base-station. Communicates with end-devices and, via IP, with a network server.

- Network Server: The Network Server (NS) terminates the LoRaWAN MAC layer for the end-devices connected to the network. It is the center of the star topology.

- Uplink message: refers to communications from end-device to network server or appliction via one or more gateways.

- Downlink message: refers to communications from network server or application via one gateway to a single end-device or a group of end-devices (considering multicasting).

- Application: refers to application layer code both on the end-device and running "behind" the network server. For LoRaWAN, there will generally only be one application running on most end-devices. Interfaces between the network server and application are not further described here.

- Classes A, B and C define different device capabilities and modes of operation for end-devices. End-devices can transmit uplink messages at any time in any mode of operation (so long as e.g., ISM band restrictions are honoured). An end-device in Class A can only receive downlink messages at predetermined timeslots after each uplink message transmission. Class B allows the end-device to receive downlink messages at periodically scheduled timeslots. Class C allows receipt of downlink messages at anytime. Class selection is based on the end-devices' application use case and its power supply. (While Classes B and C are not further described here, readers may have seen those terms elsewhere so we include them for clarity.)

3. Radio Spectrum

LoRaWAN radios make use of ISM bands, for example, 433MHz and 868MHz within the European Union and 915MHz in the Americas.

The end-device changes channel in a pseudo-random fashion for every transmission to help make the system more robust to interference and/or to conform to local regulations.

As with other LPWAN radio technologies, LoRaWAN end-devices respect the frequency, power and maximum transmit duty cycle requirements for the sub-band imposed by local regulators. In most cases, this means an end-device is only transmitting for 1% of the time, as specified by ISM band regulations. And in some cases the LoRaWAN specification calls for end-devices to transmit less often than is called for by the ISM band regulations in order to avoid congestion.

Figure 2 below shows that after a transmission slot a Class A device turns on its receiver for two short receive windows that are offset from the end of the transmission window. The frequencies and data rate chosen for the first of these receive windows match those used for the transmit window. The frequency and data-rate for the second receive window are configurable. If a downlink message preamble is detected during a receive window, then the end-device keeps the radio on in order to receive the frame.

End-devices can only transmit a subsequent uplink frame after the end of the associated receive windows. When a device joins a LoRaWAN

network (see Section 4 for details), there are similar timeouts on parts of that process.

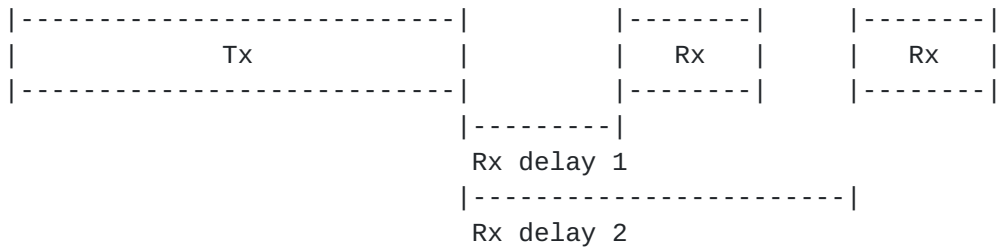

Figure 2: LoRaWAN Class A transmission and reception window

Given the different regional requirements the detailed specification for the LoRaWAN physical layer (taking up more than 30 pages of the specification) is not reproduced here. Instead and mainly to illustrate the kinds of issue encountered, in Table 1 we present some of the default settings for one ISM band (without fully explaining those here) and in Table 2 we describe maxima and minima for some parameters of interest to those defining ways to use IETF protocols over the LoRaWAN MAC layer.

Parameters	Default Value
Rx delay 1	1 s
Rx delay 2	2 s (must be RECEIVE_DELAY1 + 1s)
join delay 1	5 s
join delay 2	6 s
868MHz Default channels	3 (868.1,868.2,868.3), date rate: 0.3-5 kbps

Table 1: Default settings for EU868MHz band

Parameter/Notes	Min	Max
Duty Cycle: some but not all ISM bands impose a limit in terms of how often an end-device can transmit. In some cases LoRaWAN is more stringent in an attempt to avoid congestion.	1%	no-limit
EU 868MHz band data rate/frame-size	250 bits/s : 59 octets	50000 bits/s : 250 octets
US 915MHz band data rate/frame-size	980 bits/s : 19 octets	21900 bits/s : 250 octets

Table 2: Minima and Maxima for various LoRaWAN Parameters

Note that in the case of the smallest frame size (19 octets), 8 octets are required for LoRa MAC layer headers leaving only 11 octets for payload (including MAC layer options). However, those settings do not apply for the join procedure - end-devices are required to use a channel that can send the 23 byte Join-request message for the join procedure.

4. MAC Layer

Uplink and downlink higher layer data is carried in a MACPayload. There is a concept of "ports" (an optional 8 bit value) to handle different applications on an end-device. Port zero is reserved for LoRaWAN specific messaging, such as the join procedure.

The header also distinguishes the uplink/downlink directions and whether or not an acknowledgement ("confirmation") is required from the peer.

All payloads are encrypted and ciphertexts are protected with a cryptographic Message Integrity Check (MIC) - see Section 6 for details.

In addition to carrying higher layer PDUs there are Join-Request and Join-Response (aka Join-Accept) messages for handling network access. And so-called "MAC commands" (see below) up to 15 bytes long can be piggybacked in an options field ("FOpts").

LoRaWAN end-devices can choose various different data rates from a menu of available rates (dependent on the frequencies in use). It is however, recommended that end-devices set the Adaptive Data Rate ("ADR") bit in the MAC layer which is a signal that the network should control the data rate (via MAC commands to the end-device). The network can also assert the ADR bit and control data rates at it's discretion. The goal is to ensure minimal on-time for radios whilst increasing throughput and reliability when possible. Other things being equal, the effect should be that end-devices closer to a gateway can successfully use higher data rates, whereas end-devices further from all gateways still receive connectivity though at a lower data rate.

Data rate changes can be validated via a scheme of acks from the network with a fall-back to lower rates in the event that downlink acks go missing.

There are 16 (or 32) bit frame counters maintained in each direction that are incremented on each transmission (but not re-transmissions) that are not re-used for a given key. When the device supports a 32 bit counter, then only the least significant 16 bits are sent in the MAC header, but all 32 bits are used in cryptographic operations. (If an end-device only supports a 16 bit counter internally, then the topmost 16 bits are set to zero.)

There are a number of MAC commands for: Link and device status checking, ADR and duty-cycle negotiation, managing the RX windows and radio channel settings. For example, the link check response message allows the network server (in response to a request from an end-device) to inform an end-device about the signal attenuation seen most recently at a gateway, and to also tell the end-device how many gateways received the corresponding link request MAC command.

Some MAC commands are initiated by the network server. For example, one command allows the network server to ask an end-device to reduce it's duty-cycle to only use a proportion of the maximum allowed in a region. Another allows the network server to query the end-device's power status with the response from the end-device specifying whether it has an external power source or is battery powered (in which case a relative battery level is also sent to the network server).

The network server can also inform an end-device about channel assignments (mid-point frequencies and data rates). Of course, these must also remain within the bands assigned by local regulation.

5. Names and Addressing

A LoRaWAN network has a short network identifier ("NwkID") which is a seven bit value. A private network (common for LoRaWAN) can use the value zero. If a network wishes to support "foreign" end-devices then the NwkID needs to be registered with the LoRA Alliance, in which case the NwkID is the seven least significant bits of a registered 24-bit NetID. (Note however, that the methods for "roaming" are currently being enhanced within the LoRA Alliance, so the situation here is somewhat fluid.)

In order to operate nominally on a LoRaWAN network, a device needs a 32-bit device address, which is the catentation of the NwkID and a 25-bit device-specific network address that is assigned when the device "joins" the network (see below for the join procedure) or that is pre-provisioned into the device.

End-devices are assumed to work with one or a quite limited number of applications, which matches most LoRaWAN use-cases. The applications are identified by a 64-bit AppEUI, which is assumed to be a registered IEEE EUI64 value.

In addition, a device needs to have two symmetric session keys, one for protecting network artefacts (port=0), the NwkSKey, and another for protecting appliction layer traffic, the AppSKey. Both keys are used for 128 bit AES cryptpgraphic operations. (See Section 6 for details.)

So, one option is for an end-device to have all of the above, plus channel information, somehow (pre-)provisioned, in which case the end-device can simply start transmitting. This is achievable in many cases via out-of-band means given the nature of LoRaWAN networks. Table 3 summarises these values.

Value	Description
DevAddr	DevAddr (32-bits) = NwkId (7-bits) + device-specific network address (25 bits)
AppEUI	IEEE EUI64 naming the application
NwkSKey	128 bit network session key for use with AES
AppSKey	128 bit application session key for use with AES

Table 3: Values required for nominal operation

As an alternative, end-devices can use the LoRaWAN join procedure in order to setup some of these values and dynamically gain access to the network.

To use the join procedure, an end-device must still know the AppEUI. In addition to the AppEUI, end-devices using the join procedure need to also know a different (long-term) symmetric key that is bound to the AppEUI - this is the application key (AppKey), and is distinct from the application session key (AppSKey). The AppKey is required to be specific to the device, that is, each end-device should have a different AppKey value. And finally the end-device also needs a long-term identifier for itself, syntactically also an EUI-64, and known as the device EUI or DevEUI. Table 4 summarises these values.

Value	Description
DevEUI	IEEE EUI64 naming the device
AppEUI	IEEE EUI64 naming the application
AppKey	128 bit long term application key for use with AES

Table 4: Values required for join procedure

The join procedure involves a special exchange where the end-device asserts the AppEUI and DevEUI (integrity protected with the long-term AppKey, but not encrypted) in a Join-request uplink message. This is then routed to the network server which interacts with an entity that knows that AppKey to verify the Join-request. All going well, a Join-accept downlink message is returned from the network server to the end-device that specifies the 24-bit NetID, 32-bit DevAddr and channel information and from which the AppSKey and NwkSKey can be derived based on knowledge of the AppKey. This provides the end-device with all the values listed in Table 3.

There is some special handling related to which channels to use and for multiple transmissions for the join-request which is intended to ensure a successful join in as many cases as possible. Join-request and Join-accept messages also include some random values (nonces) to both provide some replay protection and to help ensure the session keys are unique per run of the join procedure. If a Join-request fails validation, then no Join-accept message (indeed no message at all) is returned to the end-device. For example, if an end-device is factory-reset then it should end up in a state in which it can re-do the join procedure.

6. Security Considerations

In this section we describe the use of cryptography in LoRaWAN. This section is not intended as a full specification but to be sufficient so that future IETF specifications can encompass the required security considerations. The emphasis is on describing the externally visible characteristics of LoRaWAN.

6.1. Payload Encryption and Data Integrity

All payloads are encrypted and have data integrity. Frame options (used for MAC commands) when sent as a payload (port zero) are therefore protected. MAC commands piggy-backed as frame options ("FOpts") are however sent in clear. Since MAC commands may be sent as options and not only as payload, any values sent in that manner are visible to a passive attacker but are not malleable for an active attacker due to the use of the MIC.

For LoRaWAN version 1.0.x, the NWkSkey session key is used to provide data integrity between the end-device and the network server. The AppSKey is used to provide data confidentiality between the end-device and network server, or to the application "behind" the network server, depending on the implementation of the network.

All MAC layer messages have an outer 32-bit Message Integrity Code (MIC) calculated using AES-CMAC calculated over the ciphertext payload and other headers and using the NwkSkey.

Payloads are encrypted using AES-128, with a counter-mode derived from IEEE 802.15.4 using the AppSKey.

Gateways are not expected to be provided with the AppSKey or NwkSKey, all of the infrastructure-side cryptography happens in (or "behind") the network server.

6.2. Key Derivation

When session keys are derived from the AppKey as a result of the join procedure the Join-accept message payload is specially handled.

The long-term AppKey is directly used to protect the Join-accept message content, but the function used is not an aes-encrypt operation, but rather an aes-decrypt operation. The justification is that this means that the end-device only needs to implement the aes-encrypt operation. (The counter mode variant used for payload decryption means the end-device doesn't need an aes-decrypt primitive.)

The Join-accept plaintext is always less than 16 bytes long, so electronic code book (ECB) mode is used for protecting Join-accept messages.

The Join-accept contains an AppNonce (a 24 bit value) that is recovered on the end-device along with the other Join-accept content (e.g. DevAddr) using the aes-encrypt operation.

Once the Join-accept payload is available to the end-device the session keys are derived from the AppKey, AppNonce and other values, again using an ECB mode aes-encrypt operation, with the plaintext input being a maximum of 16 octets.

7. IANA Considerations

There are no IANA considerations related to this memo.

8. Acknowledgements

The authors re-used some text from [I-D.vilajosana-lpwan-lora-hc]

Stephen Farrell's work on this memo was supported by the Science Foundation Ireleand funded CONNECT centre <https://connectcentre.ie/>.

9. Contributors

The following members of the LoRa Alliance reviewed this draft and contributed (much more than SF) to the definition of LoRaWAN.

Name, Affiliation, email (optional)

Chun-Yeow Yeoh, VADS LYFE SDN BHD, yeow@tmrnd.com.my

Olivier Hersent, Actility, olivier.hersent@actility.com

Dave Kjendal, Senet Inc, dkjendal@senetco.com

Paul Duffy, Cisco, paduffy@cisco.com

Joachim Ernst, Swisscom Broadcast Ltd, joachim.ernst@swisscom.com

Nicolas Sornin, Semtech, nsornin@semtech.com

Phillippe Christin, Orange, philippe.christin@orange.com

10. Informative References

[I-D.vilajosana-lpwan-lora-hc]
Vilajosana, X., Dohler, M., and A. Yegin, "Transmission of IPv6 Packets over LoRaWAN", draft-vilajosana-lpwan-lora-hc-00 (work in progress), July 2016.

[LoRaSpec]
LoRa Alliance, "LoRaWAN Specification Version V1.0.2", Nov 2016, <URL TBD>.

[LoRaSpec1.0]
LoRa Alliance, "LoRaWAN Specification Version V1.0", Jan 2015, <https://www.lora-alliance.org/portals/0/specs/LoRaWAN%20Specification%201R0.pdf>.

Authors' Addresses

Stephen Farrell
Trinity College Dublin
Dublin 2
Ireland

Phone: +353-1-896-2354
Email: stephen.farrell@cs.tcd.ie

Alper Yegin
Actility
Paris, Paris
FR

Email: alper.yegin@actility.com