AN1200.87

LoRaWAN® Device Classes

Table of Contents

List of figures

List of tables

Chapter 1. Introduction

A LoRaWAN®-based network is made up of end devices, gateways, a network server, and application servers. End devices send data to gateways (*uplinks*), and the gateways pass it on to the network server, which, in turn, passes it on to the application server as necessary.

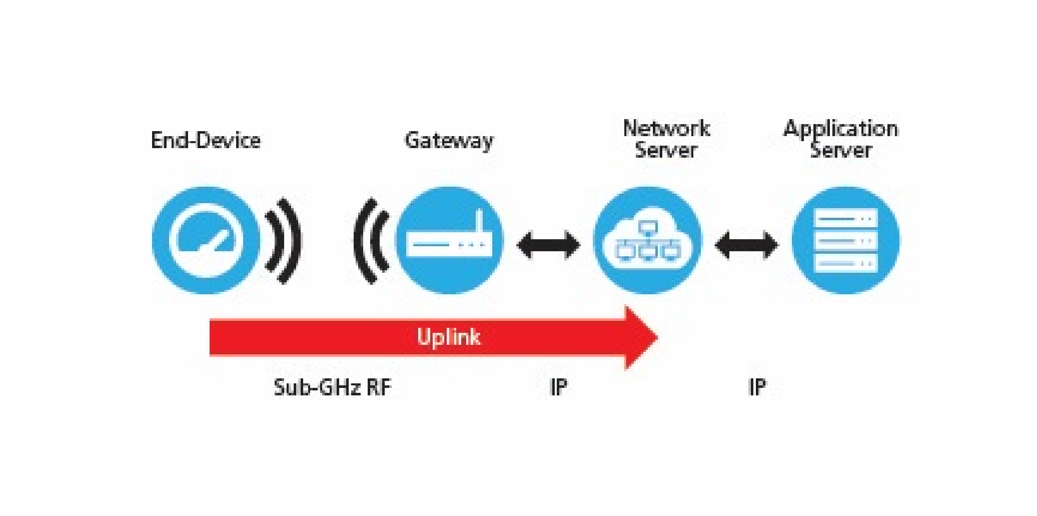


Figure 1. Uplink Transmission

Additionally, the network server can send messages (either for network management, or on behalf of the application server) through the gateways to the end devices (*downlinks*).

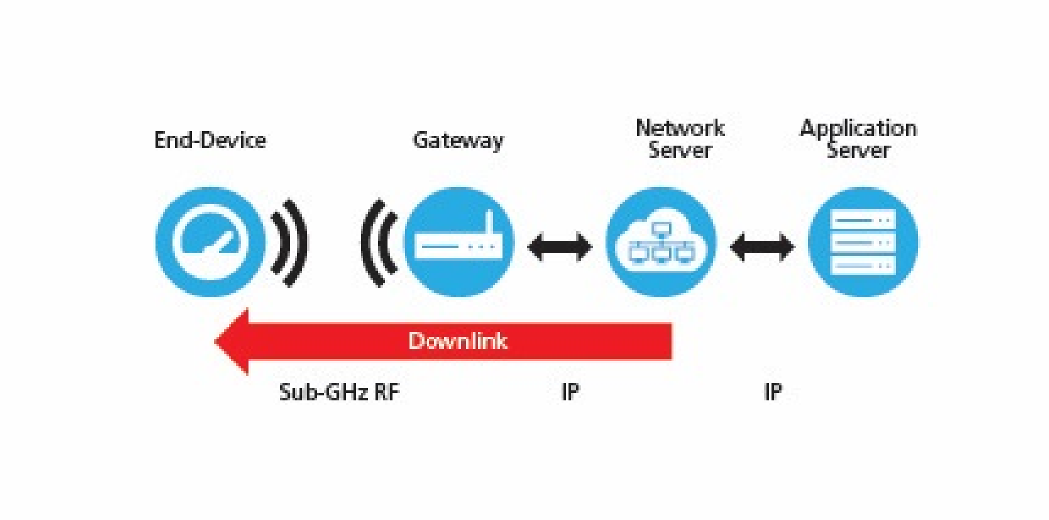


Figure 2. Downlink Transmission

End devices in a LoRaWAN network come in three classes: Class A, Class B and Class C. While end devices can always send uplinks at will, the device's class determines when it can receive downlinks. The class also determines a device's energy efficiency. The more energy efficient a device, the longer the battery life.

All end devices must support Class A ("Aloha") communications. Class A end devices spend most of their time in sleep mode. Because LoRaWAN is not a "slotted" protocol, end devices can communicate with the network server any time there is a change in a sensor reading or when a timer fires. Basically, they can wake up and talk to the server at any moment. After the device sends an uplink, it "listens" for a message from the network one and two seconds after the uplink (*receive windows*) before going back to sleep. Class A is the most energy efficient and results in the longest battery life.

In contrast, rather than only waiting for one of its sensors to notice a change in the environment or fire a timer, Class B end devices also wake up and open a receive window to listen for a downlink according to a configurable, network-defined schedule. A periodic beacon signal transmitted by the network allows those end devices to synchronize their internal clocks with the network server.

Finally, Class C ("Continuous") end devices never go to sleep. They constantly listen for downlink messages from the network, except when transmitting data in response to a sensor event. These devices are more energy-intensive, and usually require a constant power source, rather than relying on a battery.

To illustrate the different levels of power consumption for each of the different end-device classes, see Figure 3.

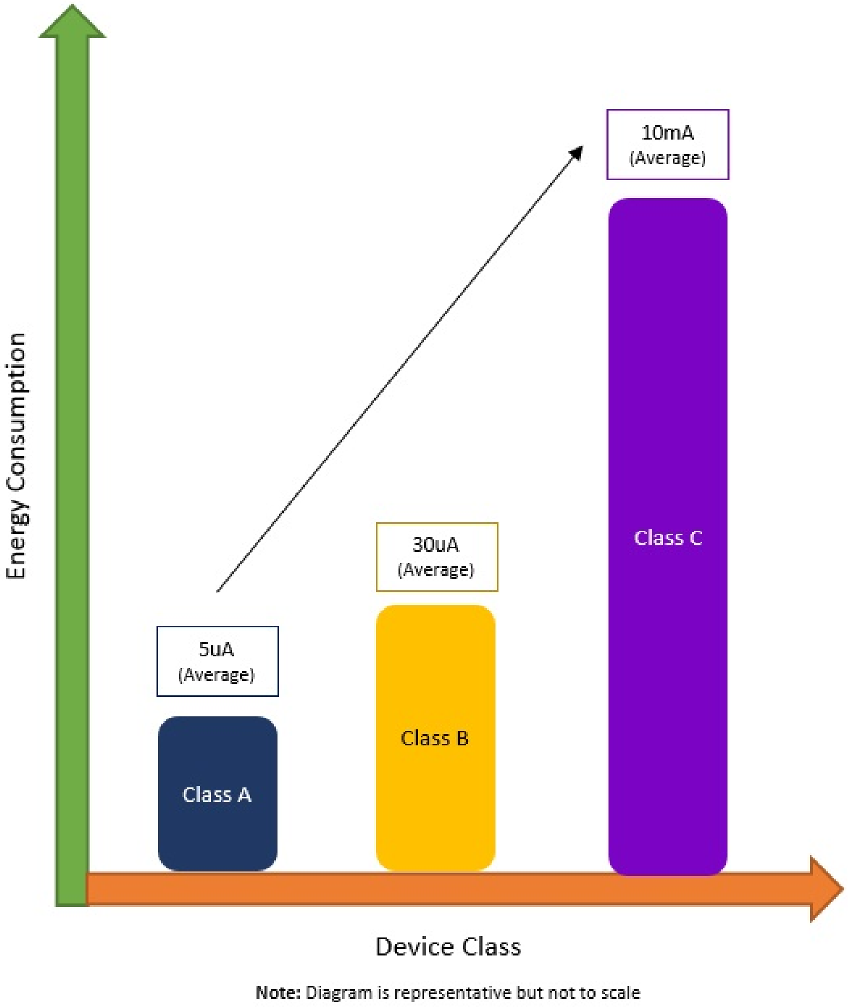


Figure 3. Energy Consumption by Device Class

In this paper, we take an in-depth look at Class A end devices.

Chapter 2. An In-depth look at LoRaWAN Class A Devices

2.1. Understanding Class A End Devices

The LoRaWAN protocol relies on an Aloha-type network. In this type of network, end devices are allowed to transmit arbitrarily.

The key characteristic of Class A is that communication is initiated only by the end device.

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.

2.1.1. Receive Windows

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. The following diagrams illustrate the different receive window state possibilities.

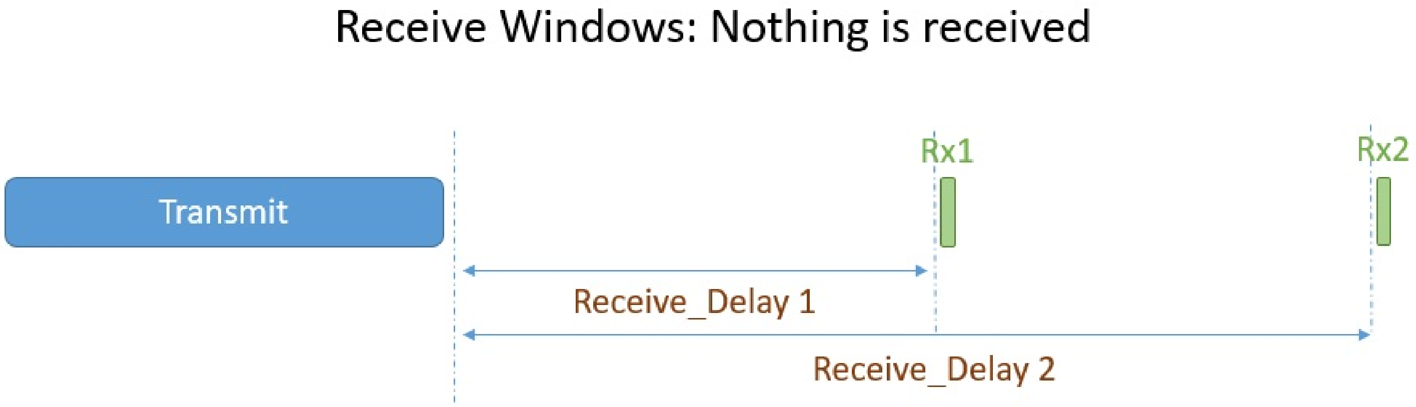


Figure 4. Nothing is received

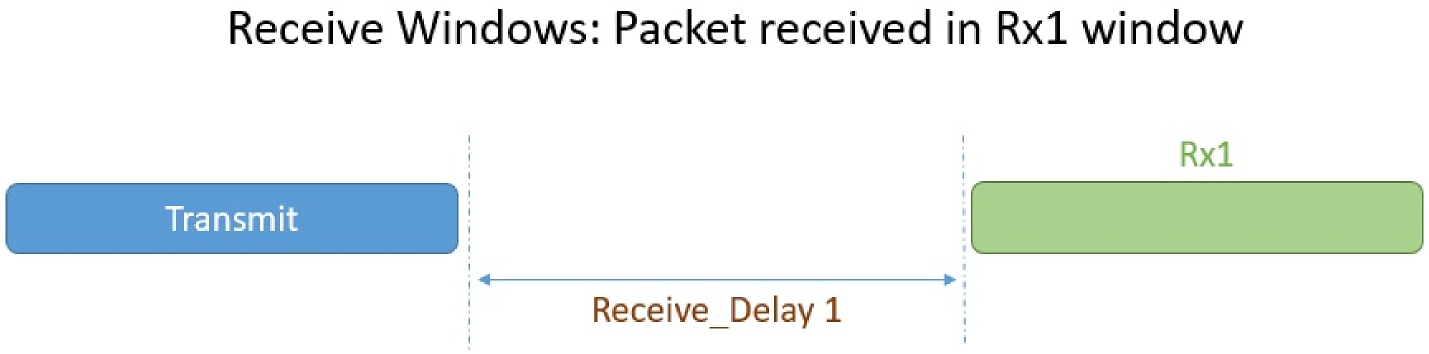


Figure 5. Packet received in Rx1 window

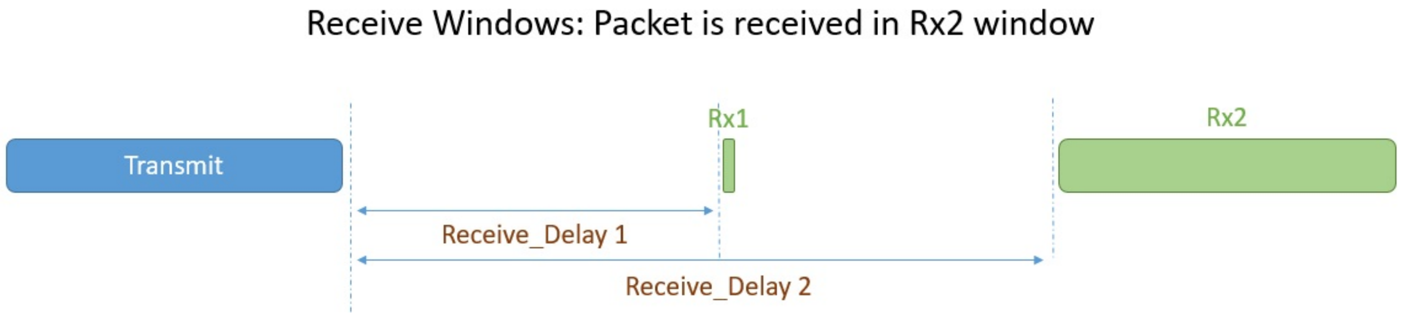


Figure 6. Packet received in Rx2 window

The default delay for Rx1 (*RECEIVE_DELAY1*) is a network parameter found in the LoRaWAN Regional Parameters document from the LoRa Alliance®. The default delay may be region-specific, and can be changed by the network operator through the MAC command *RxTimingSetupReq*. It is typically set to one second

The end device then waits one second after Rx1 closes before opening Rx2. This means that RECEIVE_DELAY2 = RECEIVE_DELAY1 + one second.

NOTE

A device will not try to send another uplink message until either:

1. It has received a downlink message during Rx1, or
2. The second receive window following the last transmission is complete

2.1.2. Transmission Frequencies and Data Rates

The frequency used for Rx1 is a function of the uplink frequency. The data rate it uses is a function of the data rate used for the uplink transmission. The default relationship between the uplink frequency and the Rx1 downlink frequency and data rate are defined in the LoRaWAN Regional Parameters document. The default parameters can be remotely reconfigured by the network operator using the relevant LoRaWAN MAC commands.

Rx2 uses a frequency and data rate that can be configured using a MAC command. The defaults are region-specific.

2.1.3. MAC Commands

With respect to Rx1, the offset between the uplink transmission (Tx) data rate and the downlink data rate can be configured using the *Rx1DRoffset* field of the *RxParamSetupReq* MAC command

To configure the Rx2 data rate, use the MAC command *RxParamSetupReq*.

The *DLChannelReq* MAC command allows the network to associate a different downlink frequency with the Rx1 window. This command works for all regions that support the *NewChannelReq* MAC command. For example, *DLChannelReq* applies in the EU and in China, but not in the U.S. or Australia, as described in the LoRaWAN Regional Parameters document.

2.2. Class A Energy Profile

The duration of each receive window must be at least as long as the time required by the end device's radio transceiver to effectively detect a downlink preamble. If the device detects a downlink preamble during this time, the radio receiver will stay open until the downlink data is demodulated.

If a downlink was detected and demodulated during Rx1 and if, (after the address and Message Integrity Check (MIC)) it is determined to be intended for the end device that received it, the device will not open Rx2 in an attempt to conserve energy. However, if the end device does not receive a downlink message during Rx1, it will continue to open Rx2 on schedule.

When there is nothing for an end device to receive, it will still open its reception windows, but just long enough to determine whether there is a preamble. At minimum, this is 5.1 milliseconds (ms) at a data rate of SF7. At maximum, it is 164 ms at a data rate of SF12. If, on the other hand, there *is* a downlink coming to the device, at SF7, it will take less than 100 ms to demodulate the message. At SF12 it can take more than two seconds.

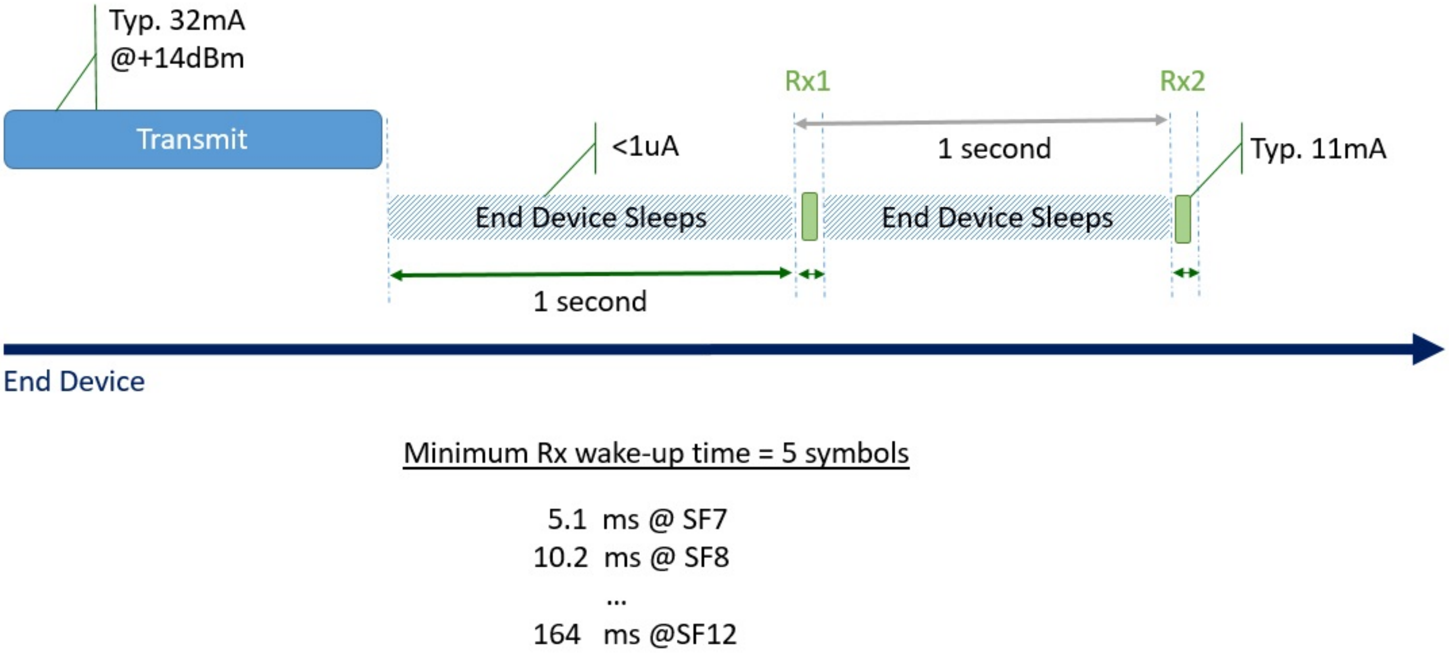


Figure 7. Uplinks, Downlinks and Wake-up Times for SX126x Radios

The energy drain linked to Rx1 and Rx2 when no downlink message is detected is negligible compared to the energy required for the transmission of the uplink

2.3. LoRaWAN Network Topology

Figure 8 illustrates the LoRaWAN network topology. The key difference between the LoRaWAN approach and others is that end devices are paired with the network itself and are not exclusively tied to a single gateway. Rather, end devices broadcast their signals to all gateways within range. Each of the receiving gateways pass the data packet along to the network server, and then the network server de-duplicates the message and sends a single version of it to the application server.

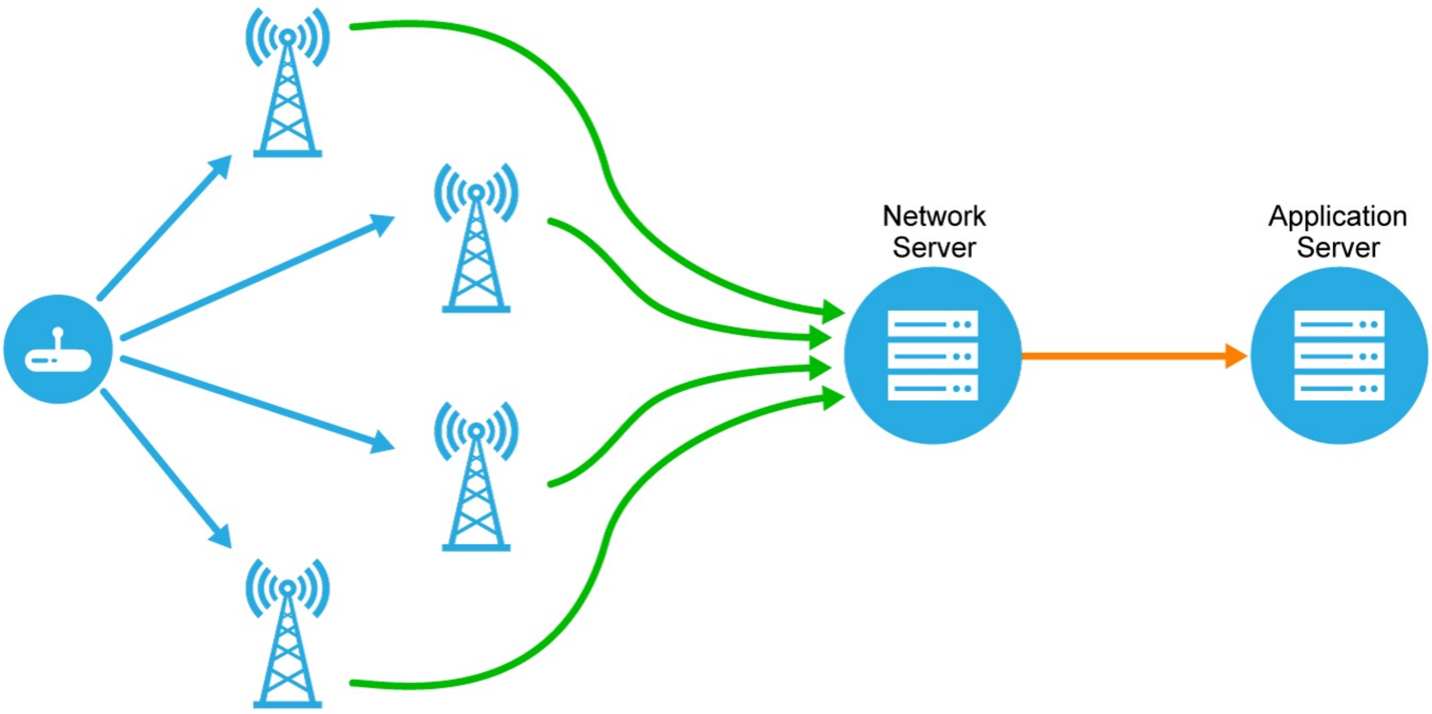


Figure 8. LoRaWAN Network Topology

There are several advantages to this topology:

- No complicated network planning is required. Gateways can be added anywhere at any time.
- Accurate message delivery is more robust, since multiple gateways receive the same data packet during each uplink. This is called uplink spatial diversity.
- There is no need to plan for different frequencies for each gateway, or to reallocate frequencies when the number of gateways change. All gateways are constantly listening to all frequencies of the network.
- Mobile devices can operate at low power thanks to the fact that any gateway can receive messages from any device. This means that (in contrast, for example, to Cellular networks) the LoRaWAN network does not notice or care that the device is moving; it simply receives uplinks from the gateways nearest the device’s current location.

2.4. Unconfirmed and Confirmed Messaging

Messages from end devices to the network and application servers and vice versa may be unconfirmed or confirmed.

When a device sends an unconfirmed message, it does not require an acknowledgement from the server. For example, most of the time a smoke detector will send periodic, unconfirmed uplinks to the network server via nearby gateways just to confirm that it is working. The gateways receive the data and pass it on to the network server, which in turn passes it on to the application server.

When sending a confirmed message, the end device requires that the message be acknowledged as received by the network server. Let’s look at our smoke detector again. When the smoke detector detects something, it will continue to transmit alerts that need to be confirmed until the alert is acknowledged. The acknowledgement informs the smoke detector that someone is responding to the alert. Because downlinks from the network are a scarce resource, they should be used sparingly. Confirmed messages should be used only for very important sensor data.

Figure 9 shows a smoke detector transmitting encrypted, unconfirmed messages (see the arrows pointing from the device at the bottom of the diagram up to the Application Server at the top via the three leftmost boxes), which are received by two gateways. The gateways add encrypted metadata to the messages and then forward them to the network server. The network server decrypts the metadata and sends the data packet to the application server, which decrypts the data.

In orange, we see the smoke detector sending an alert. These are confirmed messages, however, in the first two instances, no acknowledgement has been received, so the device continues to broadcast the alert. Finally, after the third transmission of the alert in our example below, the application server sends an acknowledgement to the device via the network server and the most appropriate gateway.

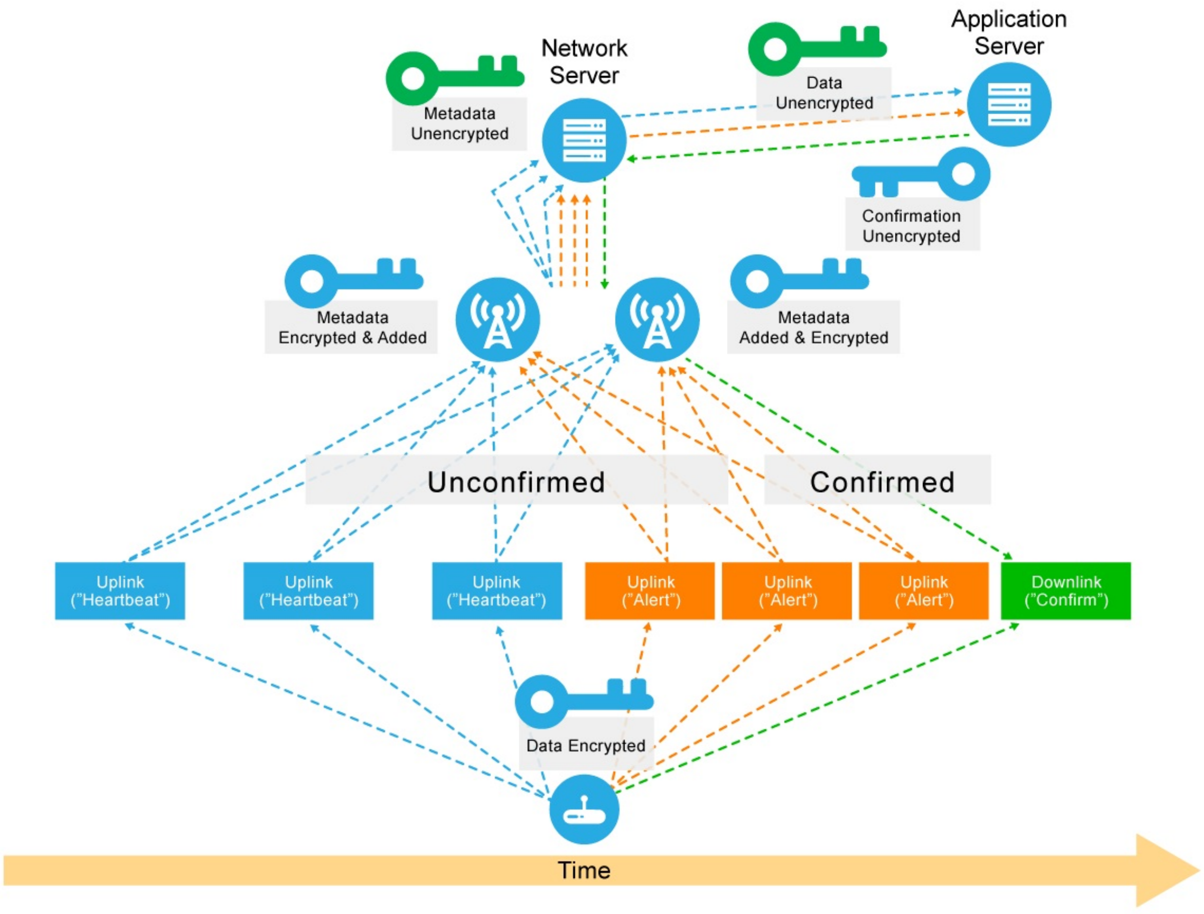


Figure 9. Unconfirmed and Confirmed Messaging

2.4.1. Application Server Downlinks

In this next example, the smoke detector application server has data to transmit to a specific, sleeping, smoke detector (the end device). However, since the smoke detector is a Class A device, the application server has to wait until the smoke detector wakes up before the server can send the data. For smoke detectors, this is a periodic message (e.g., sent every 8 hours) sharing status information—like battery status—with the application server and not a real smoke-detection event. Receipt of this message by the end device may be either confirmed or unconfirmed. Take a look at Figure 10. Here, the application server has data to send to the smoke detector, but the device is asleep. The server must wait for an uplink from the device before it can send its data. As soon as the uplink is received, the application server immediately transmits the downlink. In the case of an unconfirmed message of this sort, upon receipt of the downlink, the device goes back to sleep.

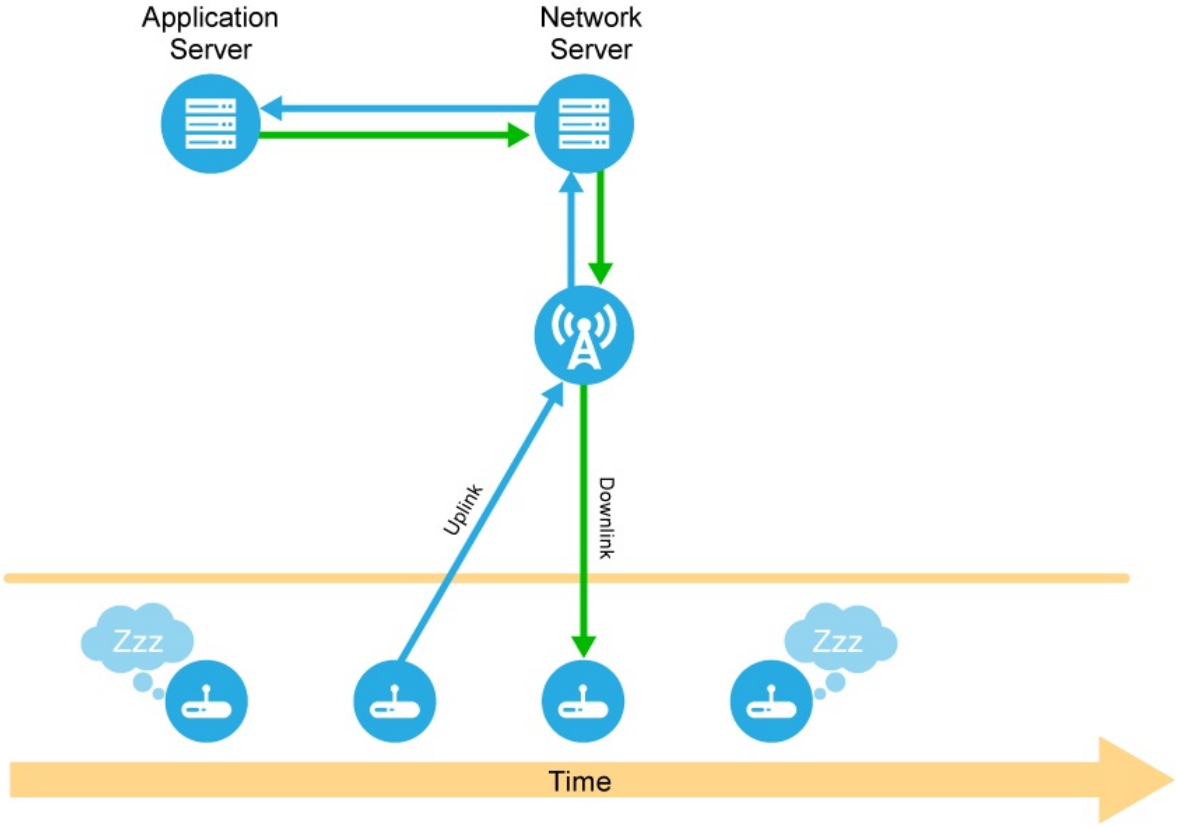


Figure 10. Application Server Downlink

2.5. Summary

In short, all end devices must support Class A operation, which is the most energy efficient mode of communication and supports the longest battery life. Class A end devices stay in deep-sleep mode until they sense a change in their environment or some other event is triggered. When this occurs, the affected device wakes up and sends its data to the network server (and eventually the application server) via one or more gateways. Following the uplink transmission, the end device will open two consecutive receive windows, during which they listen for potential downlink transmissions from the server.

Chapter 3. An In-depth Look at LoRaWAN® Class B Devices

3.1. Introduction

In a LoRaWAN® network, end devices operate in one of three modes: LoRaWAN Class A, Class B, and Class C. As described in Chapter 2 , the network can only send messages (*downlinks*) to an end device in Class A mode during one of two short receive windows, which open just after the device has sent a message (an *uplink*) to the network. These uplinks, however, are not prescheduled and may be transmitted by the device at unpredictable times. While this is great for conserving power, that power conservation comes at a cost. For example, end devices in Class A mode do not allow for a known reaction time when the customer application or the server wants to send a message to the device.

This is where LoRaWAN Class B mode can help. An enhancement of Class A, Class B mode offers regularly-scheduled, fixed-time opportunities for an end device to receive downlinks from the network. All LoRaWAN end devices start in Class A mode; however, devices programmed with a Class B stack during manufacturing may be switched to Class B mode by the application layer.

3.2. Class B: The “Beaconing” Class

End devices in Class B mode provide for regularly-scheduled receive windows, in addition to those that open whenever a Class A-style uplink is sent to the server. For this to work, a time-synchronized beacon is broadcast periodically by the network via the gateways, as illustrated in Figure 11. The end device must periodically receive one of these network beacons so that it can align its internal clock with the network.

Based on the beacon timing reference, end devices can open receive windows (*ping slots*) periodically. Any of these ping slots may be used by the network infrastructure to initiate a downlink communication.

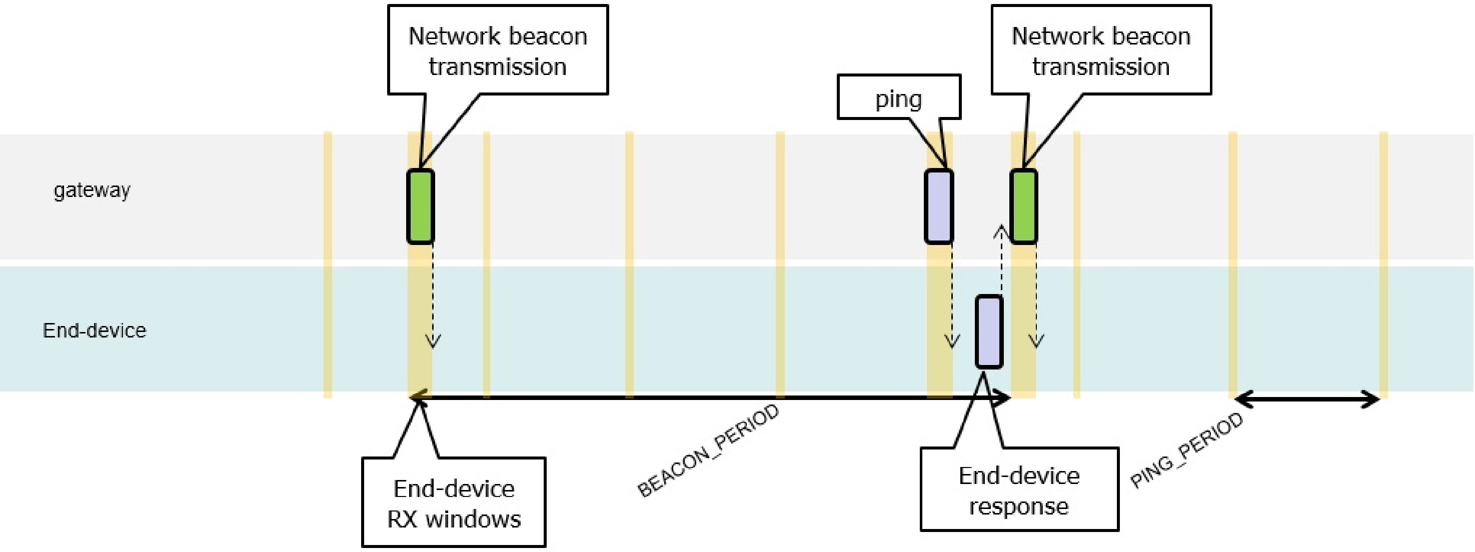


Figure 11. Network beacon opportunities

The transmission of a downlink message from an application server to a Class B device is illustrated in Figure 2. Here is a summary of the flow:

1. The application server (AS) queues a downlink message (DL) into the network server (NS)
2. The network server computes the next ping slot schedule

3. The network server computes the best gateway to use based on the last uplink received from the device and the current gateway's transmission schedule
4. The network server queues the downlink into the selected gateway
5. When the selected ping slot start time is reached, the gateway transmits the downlink
6. At the same time, the device turns its receiver on and receives the downlink

When there is no downlink to be sent (roughly 99 percent of the time), the network does not transmit anything. However, the device still opens its receiver at the beginning of the ping slot. When it does not detect a preamble, it goes back to sleep as quickly as possible to conserve power.

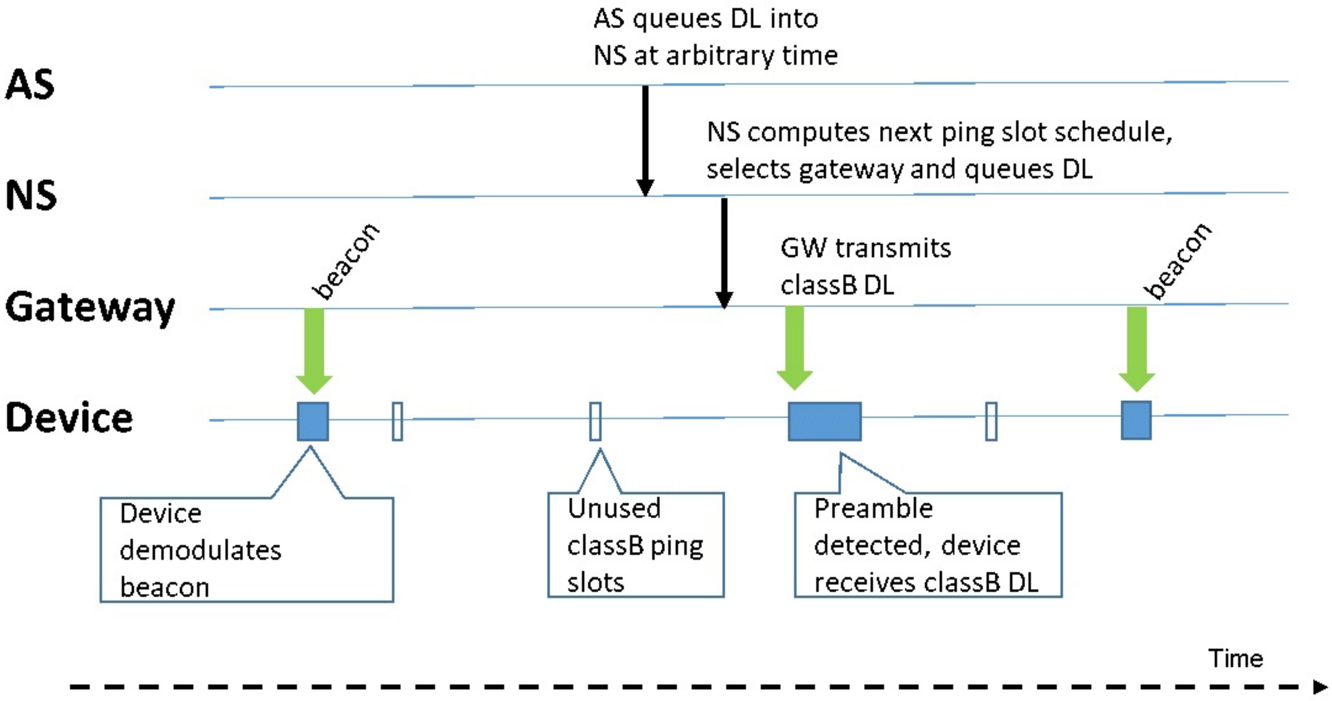

AS queues DL into NS at arbitrary time
AS
NS
NS computes next ping slot schedule, selects gateway and queues DL
Gateway
beacon
GW transmits classB DL
beacon
Device
Device demodulates beacon
Unused classB ping slots
Preamble detected, device receives classB DL
Time

Figure 12. Beacon flow

The reception of the synchronization beacons and the ping slots add additional power consumption overhead compared to a Class A device; therefore, a device is allowed to toggle between the Class A (the lowest power consumption mode) and Class B modes.

The decision to switch between the two modes is use-case specific and is left to the device's application layer.

If the switch needs to be controlled from the network side, the customer application must use one of the Class A uplinks to trigger a downlink to the application layer. The application layer on the end device must then recognize and respond to the request to switch modes—this process is not managed at the LoRaWAN level.

For a LoRaWAN network to support Class B end devices, gateways must be able to synchronously broadcast a beacon that provides a timing reference for the end devices' internal clocks. The end devices then use this timing reference to schedule when ping slots are opened to receive potential downlinks from the network.

Because all gateways broadcast the Class B beacon synchronously on the same channel and using the same radio parameters, a device within range of several gateways will receive the superposition of multiple beacons, all experiencing different attenuations and phase distortions.

In order for this end device to be able to demodulate the superposed beacons with as little interference as possible, the gateways must synchronously transmit their beacons with a timing jitter smaller than one microsecond.

This way, the various beacons that superpose at the antenna of the device actually look like a single beacon packet suffering from radio multi-path, therefore they can be demodulated.

3.3. The Beaconing Period

Synchronization beacons are transmitted by network gateways once every 128 seconds.

The interval between two consecutive beacons is divided evenly into 4096 ping slots.

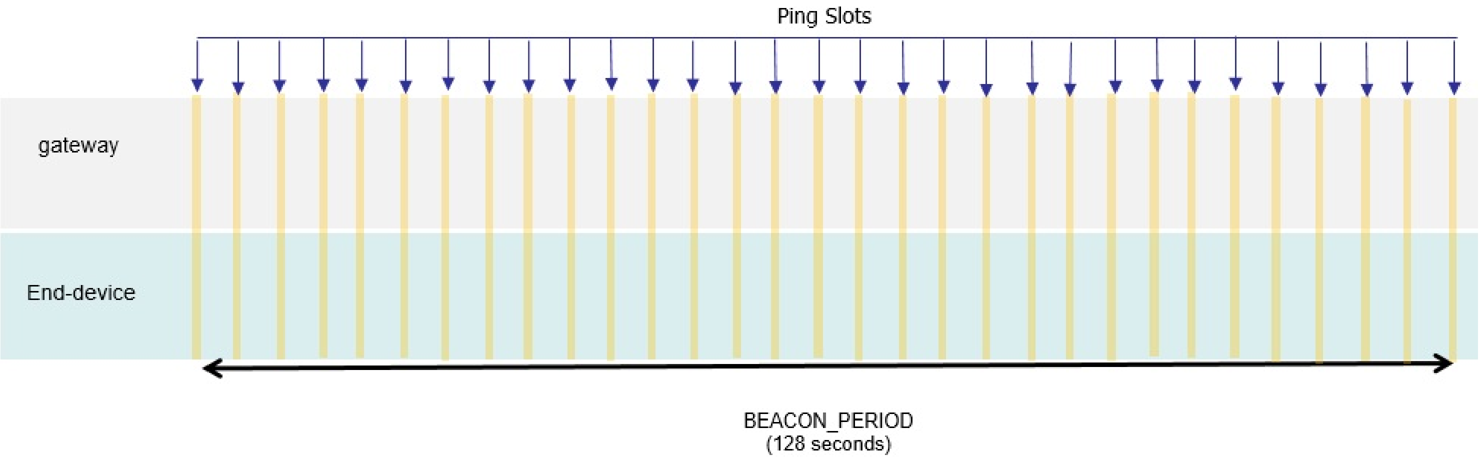


Figure 13. Ping slots

The 128-second beaconing period has been chosen as a trade-off between minimizing the gateway transmit duty-cycle (saving the gateway transmit time budget for useful downlinks from the application server), while minimizing the power consumption of the end device as it attempts to lock-on to and track the beacon. (To acquire the beacon, the device must keep its receiver on for at least one beacon period.)

You can estimate the power consumption of a Class B end device by first calculating the time on air as well as the time the receive windows are active.

3.4. Class B Power Estimation

Here is an example of how to estimate the power consumption for Class B end devices.

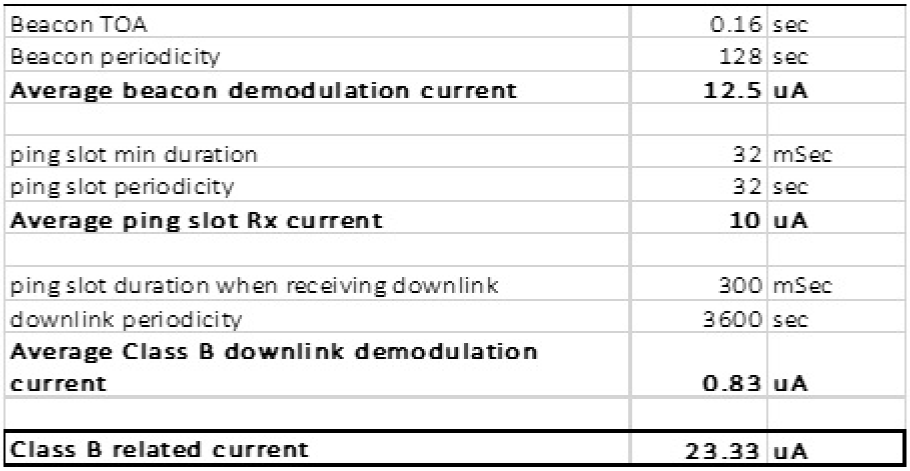
The power overhead of Class B operation is a function of:

- The time-on-air of the beacon, which is region-specific
- The periodicity of the ping slots (application driven)
- The ping slot data rate (which sets the minimum duration of each ping slot)
- The average Class B downlink periodicity

For this numerical example, we will assume the following hypothesis:

- Beacon is transmitted using SF9/125kHz beacon time-on-air ~ 160mSec
- Ping slot periodicity: 32sec (the device opens a ping slot every 16 seconds)

- Default ping slot data rate: SF9/125kHz
- Device receives, on average, one 30-byte Class B downlink per hour.
- The device current in Rx mode is 10mA (typical for SX1272/76 products)



Beacon TOA	0.16	sec
Beacon periodicity	128	sec
Average beacon demodulation current	**12.5**	**uA**
ping slot min duration	32	mSec
ping slot periodicity	32	sec
Average ping slot Rx current	**10**	**uA**
ping slot duration when receiving downlink	300	mSec
downlink periodicity	3600	sec
Average Class B downlink demodulation current	**0.83**	**uA**
Class B related current	**23.33**	**uA**

Figure 14. Estimating Class B device power consumption

We can see from this example that for a ping slot periodicity of 32 sec (meaning that the device is reachable with an average latency of 16sec), the power contribution is, roughly, split equally between the beacon synchronization and the opening of the ping slots.

The table above takes into account a possible imprecision of +/-30ppm RTC XTAL. 30ppm over 128sec leads to a +/- 4mSec maximum timing inaccuracy. The 32mSec ping slot duration is long enough to allow a +/- 4mSec misalignment between the transmitter (network) and the receiver (device). For more information about this calculation, see Application Note AN1200.23.

3.4.1. What Does the Beacon Look Like?

All beacons are transmitted using the LoRa modulation in a mode known as *implicit header* mode; that is, without a LoRa® physical header, and without CRC appended by the radio. This is possible because the beacon uses a fixed coding rate and a fixed payload length. The beacon therefore only consists of a preamble and a fixed-length payload.

The beacon preamble is longer than the default. This allows end-devices to implement a low-power, duty-cycled beacon search. This helps to minimize the power consumption of the device during beacon acquisition.

The beacon payload, *BCNPayload*, consists of three parts: a network common part (that is, this data is sent identically by all gateways), an optional gateway-specific part (this part may be different for each gateway), and a timestamp time in seconds since January 6, 1980 00:00:00 UTC (the start of the GPS epoch). The integrity of the beacon's network common part is protected by a 16-bit CRC. For specific information about how the CRC is computed, see the latest LoRaWAN specification.

The RFU field is equal to zero. (The length of the RFU field is region-specific.)

NOTE In practice, many networks only broadcast the network common part of the beacon and do not include gateway-specific fields.

3.4.2. The Gateway-Specific Beacon Element

The gateway-specific portion of the beacon payload, when present, provides additional information regarding the gateway that is sending the beacon. Therefore, this portion of the beacon will likely be different for each gateway. The gateway-specific portion of the beacon payload is protected by a CRC-16 computed on the *GwSpecific+RFU* fields. The CRC-16 definition is the same as that of the network common part.

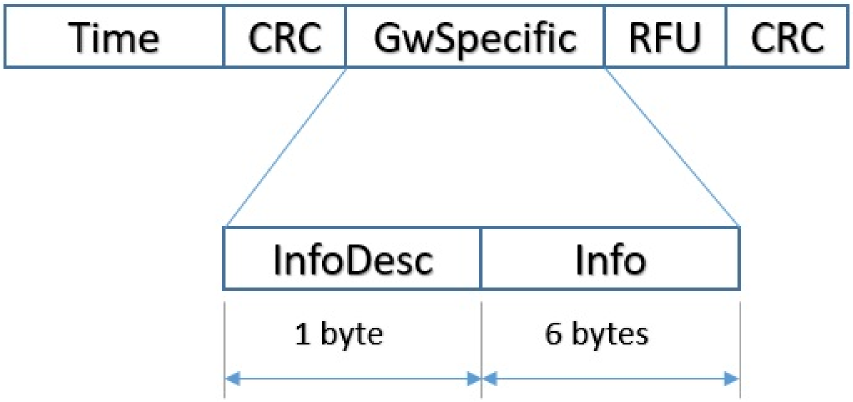


Figure 15. Beacon payload

3.4.3. The InfoDesc and Info Fields

The information descriptor, *InfoDesc*, describes how the information field should be interpreted. For a single omnidirectional antenna gateway, the *InfoDesc* value is 0 when broadcasting GPS coordinates. In contrast, for a site featuring sectored antennas for example, the first antenna broadcasts the beacon with *InfoDesc* equal to 0, and the second antenna with *InfoDesc* field equal to 1, etc. If *InfoDesc* equals 0, 1 or 2, the content of the *Info* field encodes the GPS coordinates of the antenna broadcasting the beacon.

Table 1. InfoDesc field values

InfoDesc	Description
0	GPS Coordinate of the GW’s 1st Antenna
1	GPS Coordinate of the GW’s 2nd Antenna
2	GPS Coordinate of the GW’s 3rd Antenna
3-127	Reserved Future Use
128-255	Proprietary Network Specific

The latitude and longitude (*Lat* and *Lng*) fields within the *Info* field that follows the *InfoDesc*, encode the geographical location of the gateway.

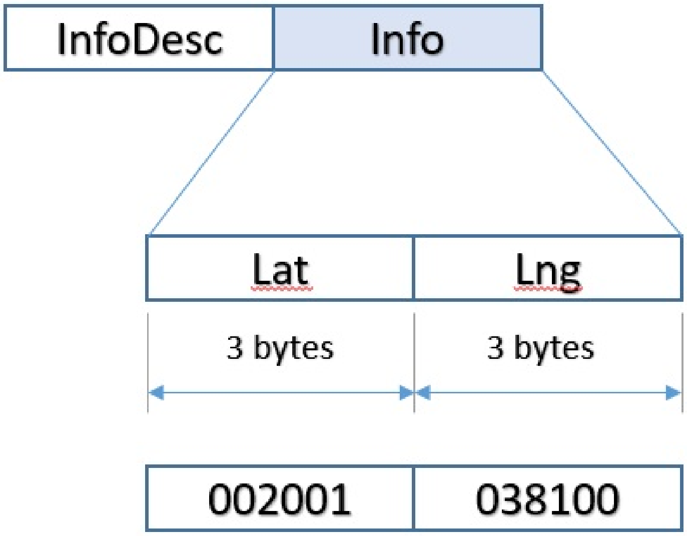


Figure 16. Info field format

The North-South latitude is encoded using a signed, 24-bit word, where -2^{23} corresponds to 90° south (the South Pole) and 2^{23} corresponds to 90° north (the North Pole). The equator corresponds to 0.

The East-West longitude is encoded using a signed 24-bit word where -2^{23} corresponds to 180° west and 2^{23} corresponds to 180° east. The Greenwich meridian corresponds to 0.

3.5. Beacon Acquisition and Tracking

As mentioned previously, an end device must receive a timing beacon to bring its internal time base in sync with the network before it can switch from Class A mode to Class B. Once successfully operating in Class B mode, the end device must stay in alignment with the network beacons. This helps ensure that the device's internal time base does not drift compared to the network time.

A device operating in Class B mode may temporarily stop receiving beacons for many reasons (interference, device moving in and out of the network coverage area, etc.).

When the device temporarily loses a beacon, it relies on its internal clock to keep opening the Class B synchronous ping slots. The LoRaWAN specification requires that Class B devices should be able to operate in Class B mode without receiving the beacon for up to two hours.

This temporary Class B operation without a beacon is called *beaconless* operation. During this holding period, the reception of any beacon directed to the end device will extend beaconless operation for another two hours.

If a Class B end device is temporarily unable to receive beacons, it will gradually widen its beacon and ping slot reception windows to account for any drift of its internal time base. Obviously, a higher precision time base allows the device to open narrower reception windows, thus minimizing its power consumption. The accuracy of the real time clock oscillator is therefore very important for Class B devices. If a Class B end device doesn't receive a beacon during the 120-minute holdover period, the end device will return to operating in Class A mode.

3.6. Ping Slots and Ping Periods

Based on the beacon timing reference, end devices periodically open *ping slots* (reception windows). A network-initiated downlink to an end device using one of these ping slots is called a *ping*. The gateway chosen to transmit this ping is selected by the network server based on signal quality indicators from the last uplink sent by the end device. For this reason, mobile Class B devices must send periodic uplinks, so that the network server can update the downlink routing-path database.

To avoid systematic collisions and problems of overhearing messages, each device's ping slot index is randomized and changed at every beacon period.

The ping slot periodicity may be as short as one second, or as long as 128 seconds, as illustrated in Figure 6.

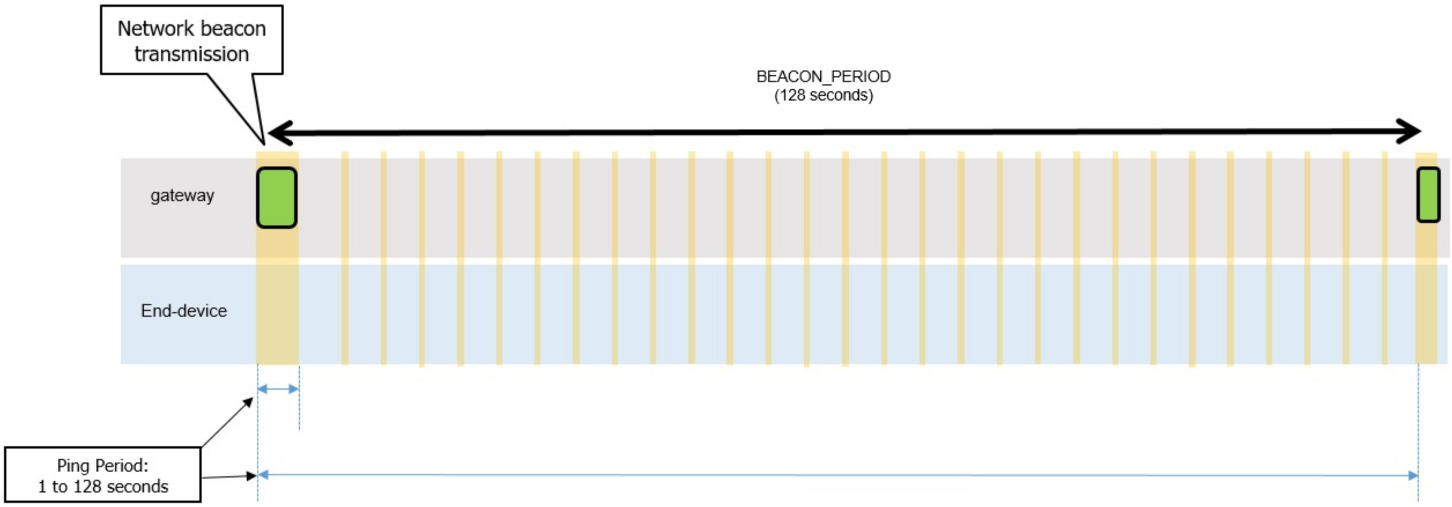


Figure 17. Ping slot periodicity

The device opens ping slots regularly. However, the network does not always have downlinks to transmit. Therefore, most ping slots are not used by the network. When this is the case, the reception window on the end device is closed as soon as the radio transceiver determines that no preamble is present. If a preamble *is* detected, the radio transceiver will remain on until the downlink frame is demodulated. The MAC layer will then process the frame, check that the address field matches the end device address, and that the Message Integrity Check (MIC) is valid. Only then will the message be forwarded to the application layer on the end device.

3.7. Unicast and Multicast Pings

Downlinks can be either *unicast* or *multicast*. Unicast messages are sent to a single end device, while multicast messages are sent to several end devices at the same time. For a multicast message to be successful, the following conditions must be met:

- All devices in a multicast group must share the same multicast address and associated encryption keys
- All devices must be open at the same time, on the same channel, using the same data rate

NOTE To remotely set up a multicast group please see the LoRaWAN Remote Multicast Setup Specification (v1.0.0)

3.8. Class B: Uplink Frame and MAC Commands

Uplink frames in Class B mode have the same structure as those used in Class A communications, with the exception of the Frame Control (*FCtrl*) octet in the frame header.

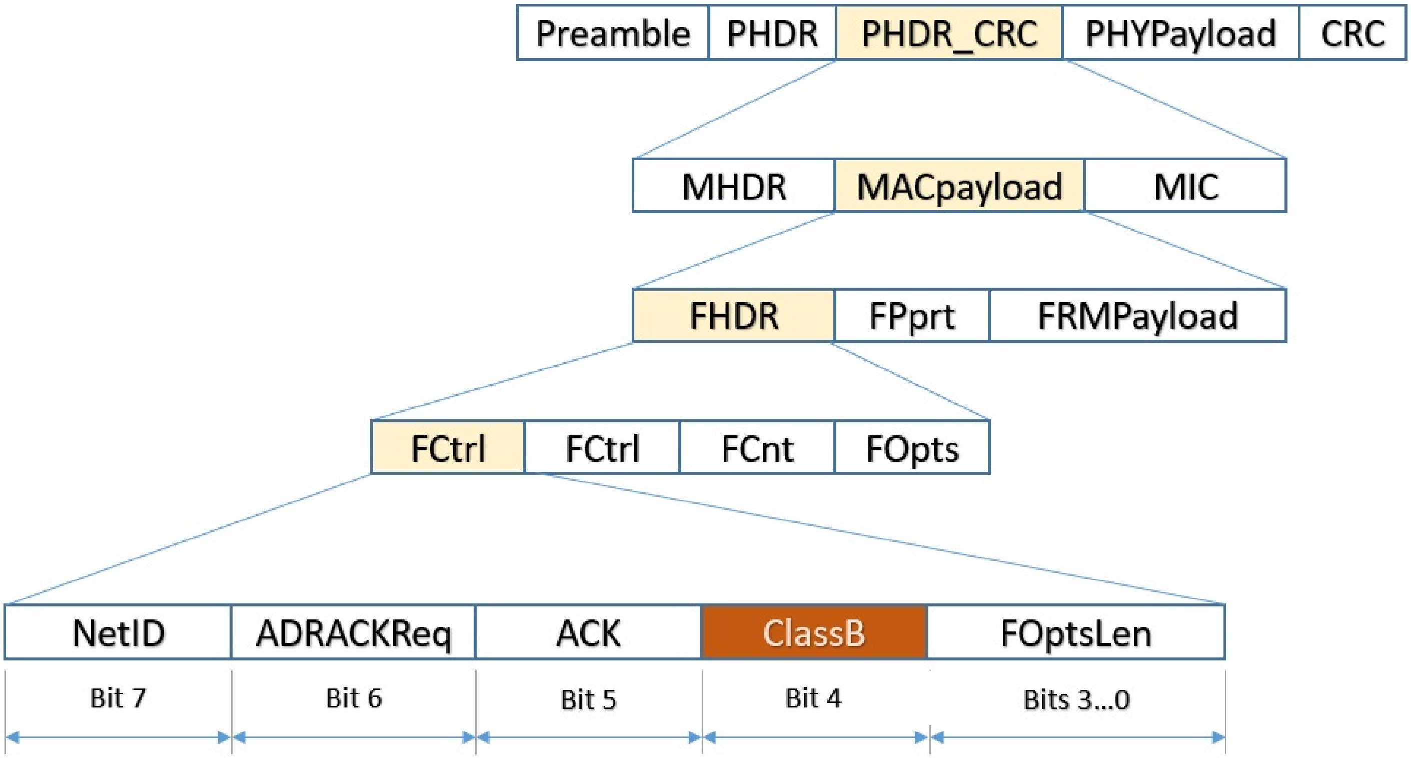


Figure 18. Uplink frame structure

For Class A communication uplinks, Bit 4 of the *FCtrl* octet is set to 0. This bit is set to 1 for Class B uplinks. Setting the Class B bit to 1 in an uplink message will signal to the network server that the device has switched to Class B mode and is ready to receive scheduled downlink pings.

Because a LoRa-based device always starts in Class A mode, All MAC commands specified for Class A end devices MUST be implemented in Class B end devices, in addition to those commands specifically used for Class B operation. For more information, see the LoRaWAN specification.

3.9. Conclusion

By design, all LoRa-based end devices support Class A communication. Class A devices can send uplinks to the network server at any time, and only listen for downlinks immediately after sending their uplinks. In contrast, Class B devices allow for predictable times when the network server can send a downlink to the device. The trade-off for this predictability is that Class B devices are not as energy efficient as those in Class A mode; therefore, the battery life of Class B devices is shorter than that of Class A devices.

Chapter 4. An In-depth Look at LoRaWAN® Class C Devices

4.1. Introduction

End devices in Class C mode are used when extremely low power consumption is not an issue and latency needs to be minimized. The server-side application determines that it is managing class C devices during the join procedure.

4.1.1. Characteristics of a Class C Device

- Based on Class A foundations
- Devices cannot simultaneously operate in Class B and Class C mode
- Lowest latency among all operating modes
- Uses more power than Class A and Class B devices

4.2. Class C: Continuous Reception

End devices operating in Class C mode have receive windows that are almost always open. These windows close only when the device is transmitting. Because of this, Class C end devices use more power to operate than Class A or Class B devices. However, in turn, they offer the lowest latency for communication from the server to an end device.

A device may be switched to Class C mode temporarily. This approach may be used to perform a firmware upgrade of a battery-powered device. A battery-powered Class A device may switch to Class C for a few minutes at a given time in order to receive a firmware update over-the-air (FUOTA) broadcast. Once the broadcast of the update is complete, the device can return to its default Class A, low-power mode of operation.

Class C end devices implement the same two receive windows as Class A devices, but they do not close the RX2 window until they send the next transmission back to the server. Therefore, they can receive a downlink in the RX2 window at almost any time. A short window at the RX2 frequency and data rate is also opened between the end of the transmission and the beginning of the RX1 receive window, as illustrated in Figure 19.

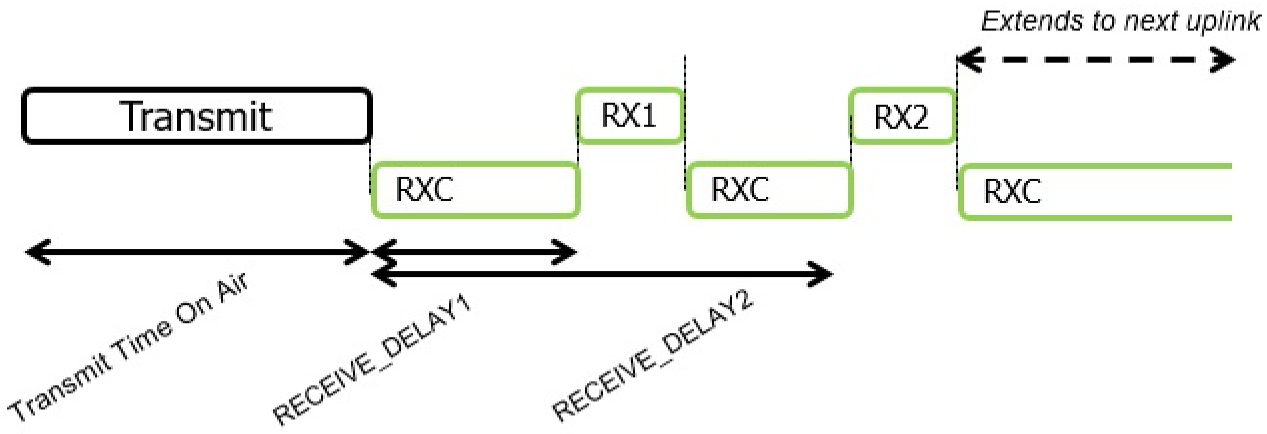


Figure 19. Receive Windows: Packet can be received in RX2 window

4.3. Multicast Downlinks

Class C end devices may receive multicast downlink frames in a manner similar to end devices in Class B mode. The multicast address and the associated network session key and application session key must come from the application layer.

The **FPending** bit is used to indicate that there is more multicast data to be sent. Given that a Class C device keeps it receiver active most of the time, the **FPending** bit does not trigger any specific behavior on the end device.

4.4. MAC Commands

All commands described in the Class A specification must be supported in Class C end devices.

For more information about other end device operating modes, see Chapter 2 and Chapter 3 .

For the full range of LoRaWAN® training offerings, please visit the Semtech Learning Center.

Disclaimer

IMPORTANT

Information relating to this product and the application or design described herein is believed to be reliable, however such information is provided as a guide only and Semtech assumes no liability for any errors in this document, or for the application or design described herein.

Semtech reserves the right to make changes to the product or this document at any time without notice. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Semtech warrants performance of its products to the specifications applicable at the time of sale, and all sales are made in accordance with Semtech's standard terms and conditions of sale.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS, OR IN NUCLEAR APPLICATIONS IN WHICH THE FAILURE COULD BE REASONABLY EXPECTED TO RESULT IN PERSONAL INJURY, LOSS OF LIFE OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

The Semtech name and logo are registered trademarks of the Semtech Corporation. All other trademarks and trade names mentioned may be marks and names of Semtech or their respective companies. Semtech reserves the right to make changes to, or discontinue any products described in this document without further notice. Semtech makes no warranty, representation or guarantee, express or implied, regarding the suitability of its products for any particular purpose. All rights reserved.