

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.4480

(11/2021)

SERIES Y: GLOBAL INFORMATION
INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS,
NEXT-GENERATION NETWORKS, INTERNET OF
THINGS AND SMART CITIES

Internet of things and smart cities and communities –
Frameworks, architectures and protocols

Low power protocol for wide area wireless networks

Recommendation ITU-T Y.4480

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE

General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899

INTERNET PROTOCOL ASPECTS

General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999

NEXT GENERATION NETWORKS

Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Computing power networks	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999

FUTURE NETWORKS

CLOUD COMPUTING

BIG DATA

QUANTUM KEY DISTRIBUTION NETWORKS

INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES

General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.4480

Low power protocol for wide area wireless networks

Summary

Recommendation Y.4480 describes a low power protocol for wide area wireless networks, which is optimized for battery-powered end-devices that may either be mobile or mounted at a fixed location.

NOTE – This protocol is technically equivalent to (and compatible with) the LoRaWAN[®] Link layer specification protocol [b-LoRaWAN TS001-1.0.4].

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4480	2021-11-29	20	11.1002/1000/14818

Keywords

LoRaWAN, long range, low power protocol, wide area wireless networks.

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2021

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

		Page
1	Scope.....	1
2	References.....	1
3	Definitions	1
	3.1 Terms defined elsewhere	1
	3.2 Terms defined in this Recommendation.....	1
4	Abbreviations and acronyms	2
5	Conventions	2
6	Introduction.....	2
	6.1 Classes of end-devices.....	4
	Class A All end-devices.....	5
7	Physical packet formats	5
	7.1 Uplink packets.....	5
	7.2 Downlink packets	5
	7.3 Receive windows.....	5
8	MAC frame formats.....	6
	8.1 PHY payload (PHYPayload).....	7
	8.2 MAC header (MHDR field).....	7
	8.3 MAC payload of data frames (MACPayload).....	8
	8.4 Message integrity code (MIC).....	14
9	MAC commands.....	15
	9.1 Link check commands (<i>LinkCheckReq</i> , <i>LinkCheckAns</i>).....	17
	9.2 Link ADR commands (<i>LinkADRReq</i> , <i>LinkADRAns</i>)	18
	9.3 End-device transmit duty cycle (<i>DutyCycleReq</i> , <i>DutyCycleAns</i>).....	20
	9.4 Receive windows parameters (<i>RXParamSetupReq</i> , <i>RXParamSetupAns</i>)	21
	9.5 End-device status (<i>DevStatusReq</i> , <i>DevStatusAns</i>).....	22
	9.6 Creation / modification of a channel (<i>NewChannelReq</i> , <i>NewChannelAns</i> , <i>DIChannelReq</i> , <i>DIChannelAns</i>)	22
	9.7 Setting delay between TX and RX (<i>RXTimingSetupReq</i> , <i>RXTimingSetupAns</i>).....	24
	9.8 End-device transmit parameters (<i>TXParamSetupReq</i> , <i>TXParamSetupAns</i>) ..	25
	9.9 End-device time commands (<i>DeviceTimeReq</i> , <i>DeviceTimeAns</i>)	26
10	End-device activation	27
	10.1 Data stored in the end-device after activation	27
	10.2 Over-the-air activation.....	28
	10.3 Activation by personalization.....	32
11	Retransmissions backoff.....	32
	Class B Beacon	34

	Page
12 Introduction to Class B	34
12.1 Principle of synchronous network-initiated Class B downlinks.....	34
13 Class B frame formats	36
13.1 Uplink frames	36
13.2 Downlink frames	36
13.3 Downlink ping frames	36
14 Class B beacon acquisition and tracking	38
14.1 Minimal beaconless operation time.....	38
14.2 Extension of beaconless operation upon receipt	38
14.3 Minimizing timing drift.....	39
15 Class B downlink slot timing.....	39
15.1 Definitions	39
15.2 Slot randomization.....	40
16 Class B MAC commands.....	41
16.1 PingSlotInfoReq	41
16.2 BeaconFreqReq	42
16.3 PingSlotChannelReq.....	43
16.4 <i>BeaconTimingReq</i> and <i>BeaconTimingAns</i>	44
17 Class B beaconing.....	44
17.1 Beacon frame format	44
17.2 Beacon <i>GwSpecific</i> field format	45
17.3 Beaconing precise timing	46
17.4 Network downlink route update requirements	47
18 Class B unicast and multicast downlink channel frequencies	47
18.1 Single-channel beacon transmission.....	47
18.2 Frequency-hopping beacon transmission	47
Class C Continuous listening end-devices.....	49
19 Introduction to Class C	49
19.1 Class C multicast downlinks	51
Appendix I Examples.....	52
I.1 Uplink timing diagram for unconfirmed data frames	52
I.2 Uplink timing diagram for confirmed data frames	53
I.3 Downlink diagram for confirmed data frames.....	54
I.4 Downlink timing for frame-pending frames	55
Bibliography.....	57

Recommendation ITU-T Y.4480

Low power protocol for wide area wireless networks

1 Scope

This Recommendation specifies a low power, wide area networking protocol designed to wirelessly connect battery operated 'things' to the internet in regional, national or global networks, and targets key internet of things requirements such as bidirectional communication, end-to-end security, mobility and localization services.

NOTE – This Recommendation is technically equivalent to [b-LoRaWAN TS001-1.0.4].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[IEEE 802.15.4] IEEE Std 802.15.4 (2011), *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*.

[IETF RFC 4493] IETF RFC 4493 (2006), *The AES-CMAC Algorithm*.

[NIST FIPS 197] NIST FIPS 197 (2001), *Advanced Encryption Standard (AES)*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following term defined elsewhere:

3.1.1 nNetwork [b-ITU-T G.993.5]: A collection of interconnected elements that provide connection services to users.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 application server: Network element that exchanges encrypted application payloads with an end-device.

3.2.2 class: Mode of operation of an end-device.

3.2.3 join procedure: Activation procedure of an end-device over a network.

3.2.4 join server: Network element that provides session keys to an application server and to a network server.

3.2.5 network server: Network element that controls the operation of end-devices and gateways.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ABP	Activation By Personalization
ADR	Adaptive Data Rate
AES	Advanced Encryption Standard
CCM	Counter with Cipher block chaining Message
CMAC	Cipher-based Message Authentication Code
CRC	Cyclic Redundancy Check
DR	Data Rate
ECB	Electronic Code Book
EIRP	Effective Isotropic Radiated Power
EUI	Extended Unique Identifier
FUOTA	Firmware Updates Over The Air
IP	Internet Protocol
MAC	Medium Access Control
MIC	Message Integrity Code
OTAA	Over-The-Air Activation
RF	Radio Frequency
RFU	Reserved for Future Usage
RX	Reception
SNR	Signal-to-Noise Ratio
TX	Transmission

5 Conventions

In the body of this Recommendation, the words "SHALL", "SHALL NOT", "SHOULD" or "IS RECOMMENDED", "MAY" and "OPTIONAL" sometimes appear in all capitals, in which case they are to be interpreted, respectively, as "is required to", "is prohibited from", "is recommended", "can optionally" and "optional". The appearance of such phrases or keywords in an appendix or a material explicitly marked as informative is to be interpreted as having no normative intent.

Medium access control (MAC) commands are written in bold italics like ***LinkCheckReq***. Bits and bit fields are written with this font: FRMPayload. Constants are written all in capitals like RECEIVE_DELAY1, and variables are written in italics like *N*.

The octet order for all multi-octet fields SHALL be little endian.

Extended unique identifiers (EUI) are 8-octet fields and SHALL be transmitted as little endian.

By default, reserved for future usage (RFU) bits SHALL be set to 0 by the transmitter of the frame and SHALL be silently ignored by the receiver.

6 Introduction

Networks as defined in this Recommendation are typically laid out in a star-of-stars topology in which gateways (also known as concentrators, routers, access points or base stations) relay transmissions

between end-devices and a central network server at the backend (see Figure 1). Gateways are connected to a network server via standard internet protocol (IP) connections, whereas end-devices use single-hop radio frequency (RF) communication to one or many gateways. All communication is generally bidirectional, although uplink communication from an end-device to a network server is expected to be the predominant traffic.

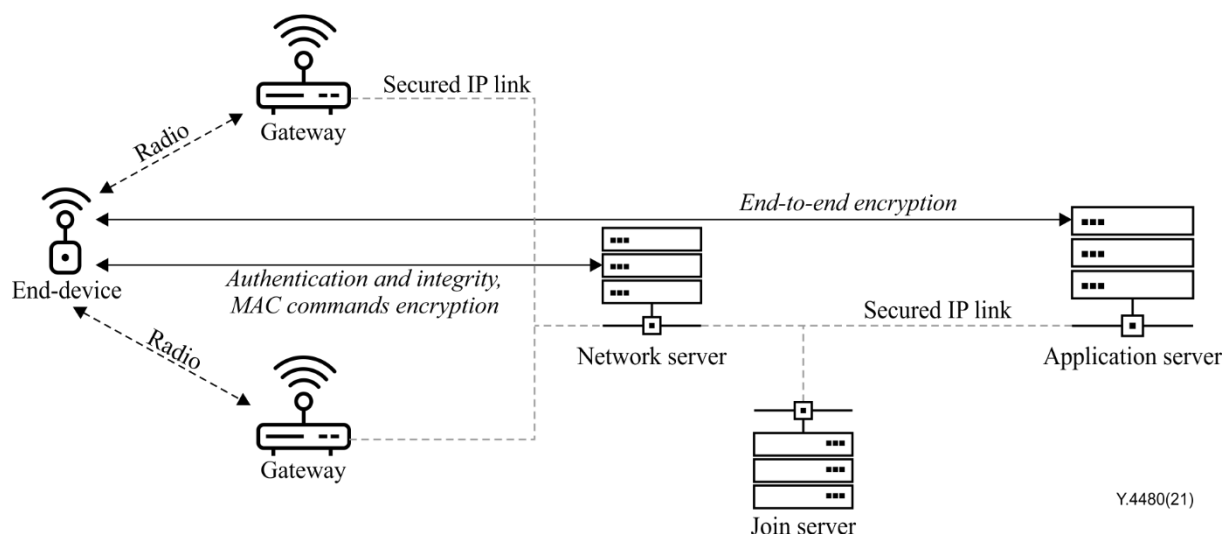


Figure 1 – Network architecture

Communication between end-devices and gateways is distributed over different frequency channels and data rates. Selecting the data rate (DR) is a trade-off between communication range and transmission (TX) duration. To maximize both the battery life of end-devices and the overall network capacity, the network infrastructure may manage the data rate and radio frequency (RF) transmit power for each end-device individually by means of an adaptive data rate (ADR) scheme.

An end-device may transmit on any channel available at any time using any available data rate, as long as the following rules are observed:

- the end-device changes channels in a pseudo-random fashion for every transmission; the resulting frequency diversity makes the system more robust to interference;
- the end-device pseudo-randomly changes its transmit periodicity to prevent systematic synchronization of populations of end-device transmissions;
- the end-device complies with all local regulations governing its behaviour in the band and sub-bands in which it is currently operating including, but not limited to, duty-cycle and dwell-time (transmit-duration) limitations.

All end-devices shall implement at least Class A functionality as described in this Recommendation. In addition, they may implement Class B and/or Class C also described in this Recommendation.

End-devices implementing Class B are referred to as Class B-capable. When operating in Class B, end-devices are referred to as Class B-enabled. Transition from Class B-disabled to Class B-enabled is called switching to Class B.

End-devices implementing Class C are referred to as Class C-capable. When operating in Class C, end-devices are referred to as Class C-enabled. Transition from Class C-disabled to Class C-enabled is called switching to Class C.

In all cases, end-devices remain compatible with Class A.

6.1 Classes of end-devices

A network distinguishes between a basic protocol (called Class A) and optional features (Class B, Class C...) as shown in Figure 2.

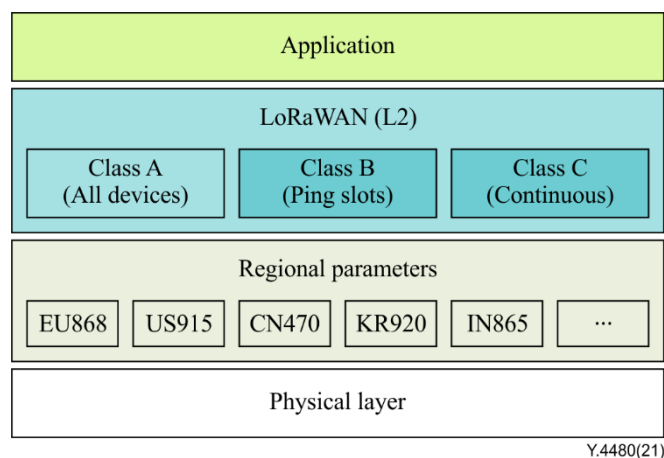


Figure 2 – Classes of end-devices

- **Bidirectional end-devices (Class A):** Class A end-devices allow bidirectional communications, whereby each end-device's uplink transmission is followed by two short downlink receive windows. The transmission slot scheduled by the end-device is based on its own communication needs, with a small variation based on a random time basis. Class A operation is the lowest power end-device system for applications that require only downlink communication from the server, shortly after the end-device has sent an uplink transmission. Downlink communications from the server at any other time will have to wait until the next uplink is initiated by the end-device.
- **Bidirectional end-devices with scheduled receive slots (Class B):** Class B-capable end-devices allow more receive slots. In addition to Class A receive windows, Class B-enabled end-devices open extra receive windows at scheduled times. In order for the end-device to open its receive windows at a scheduled time, it receives a time-synchronized beacon from the gateway. This allows the network server to know when the end-device is listening.
- **Bidirectional end-devices with maximal receive slots (Class C):** Class C-capable end-devices allow nearly continuously open receive windows, which are closed only when transmitting. Class C-enabled end-devices use more power to operate than Class A or Class B-enabled, but they feature the lowest latency for communication between servers and end-devices.

Class A

All end-devices

All end-devices SHALL implement all Class A features not explicitly marked optional.

NOTE – Physical packet format, MAC frame format and other parts of this Recommendation that are common to both end-devices of Class A and higher Classes (i.e., end-devices implementing more than Class A) are described only in the Class A specification in order to avoid redundancy.

7 Physical packet formats

The protocol defined in this Recommendation may use any long range physical layer suitable to star the network topology. If an unlicensed spectrum is used, local regulations apply. The protocol has parameters which depend on local regulations and a set of such parameters is referred to as a regional channel plan or region. This Recommendation can handle fixed channel plans where the list of usable channels is fixed, and dynamic channel plans where, dynamically created channels may be added to the default channels. An example of possible channel plans and physical layer mappings, defined for several regulatory regions, may be found in [b-LoRaWAN RP002].

This Recommendation distinguishes between uplink and downlink frames.

7.1 Uplink packets

Uplink packets are sent by end-devices to a network server relayed by one or many gateways.

7.2 Downlink packets

Downlink packets are sent by a network server to only one end-device and are transmitted by a network server through one or more gateways.

NOTE – This Recommendation does not describe the transmission of multicast frames from a network server to many end-devices.

7.3 Receive windows

Following each uplink transmission, the end-device SHALL open one or two receive windows (RX1 and RX2); if no packet destined for the end-device is received in RX1, it SHALL open RX2. The receive windows start times are defined using the end of the transmission as a reference (see Figure 3).

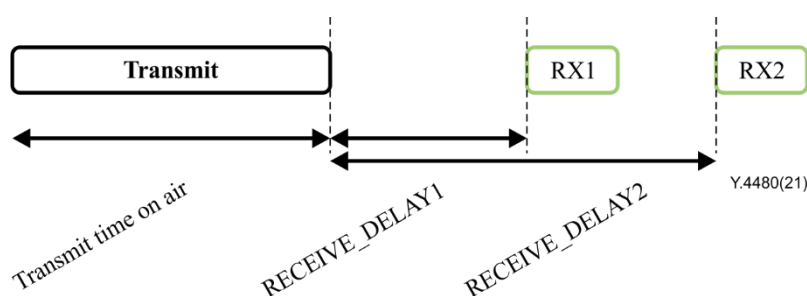


Figure 3 – End-device receive-slot timing

7.3.1 Receiver activity during receive windows

If a preamble is detected during one of the receive windows, the radio receiver SHOULD stay active until the downlink frame is demodulated. If a frame was detected and subsequently demodulated during the first receive window, and the frame was intended for this end-device after the address and message integrity code (MIC) checks, the end-device SHALL NOT open the second receive window.

7.3.2 First receive-window channel, data rate, and start

The first receive window RX1 uses a frequency that is a function of the uplink frequency and a data rate that is a function of the uplink data rate. RX1 SHALL be open no longer than RECEIVE_DELAY1 seconds after the end of the uplink modulation. The relationship between uplink and RX1 physical layer data rates vary and examples of region-specific definitions may be found in the regional parameters [b-LoRaWAN RP002].

NOTE – RECEIVE_DELAY1 is described in clause 10.

7.3.3 Second receive window channel, data rate, and start

The second receive window RX2, if opened (see clause 7.3.1), uses a fixed configurable frequency and data rate, and SHALL be open no longer than RECEIVE_DELAY2 seconds after the end of the uplink modulation. The frequency and data rate can be modified by MAC commands (see clause 9). The default frequency and data rate depend on the physical layer and the regulatory region.

NOTE – RECEIVE_DELAY2 is described in clause 10.

7.3.4 Receive window duration

The duration of a receive window SHALL be at least the time required by the end-device's radio transceiver to detect a downlink preamble starting at RECEIVE_DELAY1 or RECEIVE_DELAY2 after the end of the uplink modulation.

NOTE – The end-device receive window duration has to accommodate the maximum potential imprecision of the end-device's clock. The delay between the end of the uplink and the start of the receive windows can be changed from 1 s to 15 s for RX1 (and thereby from 2 s to 16 s for RX2) using the over-the-air activation (OTAA) Join-Accept frame or the **RXTimingSetupReq** MAC command. Therefore, this delay must be accommodated when computing the maximum clock imprecision.

EXAMPLE – A 30 ppm XTAL frequency error translates to $\pm 30 \mu\text{s}$ after 1 s and $\pm 450 \mu\text{s}$ after 15 s.

7.3.5 Network transmitting to an end-device

If a network server intends to transmit a downlink to an end-device, it SHALL initiate the transmission of the downlink frame precisely at the beginning of one of the two receive windows. Such a downlink is referred to as a Class A downlink. The end-device SHALL open a Class A RX1 receive window. If no frame intended for this end-device was received during the RX1 receive window, the end-device SHALL open an RX2 receive window at the specified timing, even if this interrupts the reception of a transmission using Class B or Class C downlink timing and receive parameters.

7.3.6 Important notice regarding receive windows

An end-device SHALL NOT transmit another uplink packet before it has either received a downlink packet in the first or second receive window related to the previous transmission or if the second receive window related to the previous transmission has expired.

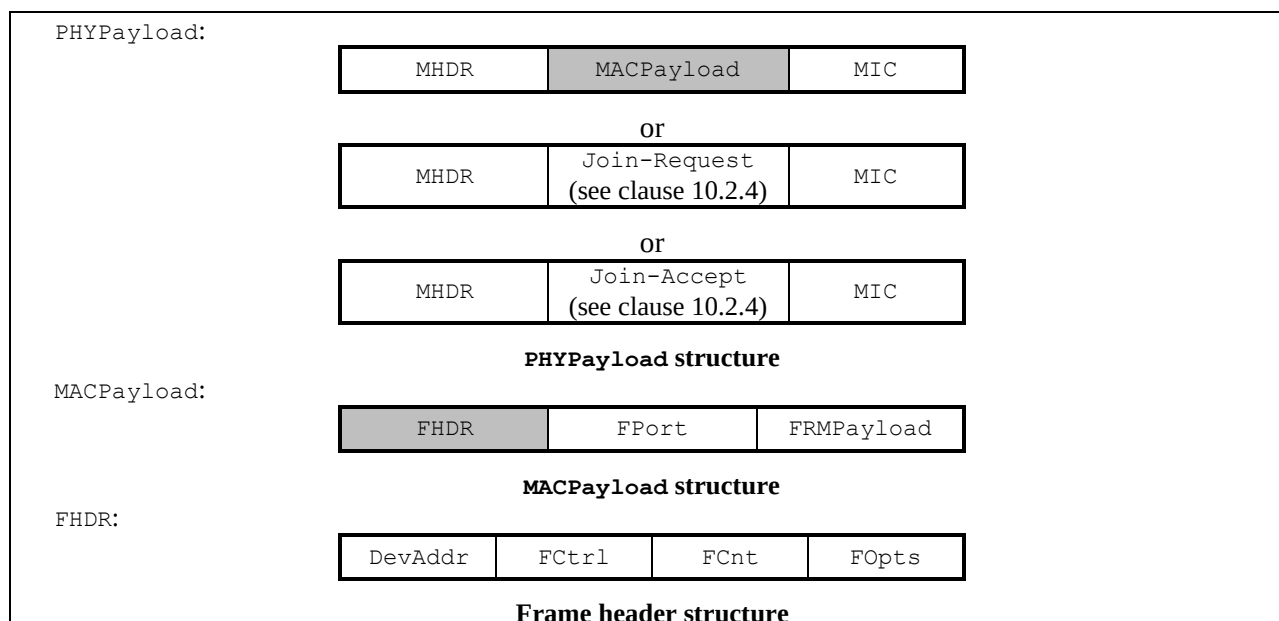
7.3.7 Receiving or transmitting other protocols

The end-device MAY listen for or transmit other protocols or perform arbitrary processing between transmission and reception windows, as long as the end-device remains compatible with local regulations and compliant with this Recommendation.

8 MAC frame formats

All uplink and downlink packets carry a PHY payload (PHYPayload) starting with a single-octet MAC header (MHDR), followed by a MAC payload (MACPayload), and ending with a 4-octet MIC, as shown in Table 1.

Table 1 – Frame format elements



8.1 PHY payload (PHYPayload)

The maximum length M of the MACPayload field depends on the physical layer and regulatory region (see Table 2). A region and data-rate specific example may be found in [b-LoRaWAN RP002]. Neither the end-device nor the network SHALL send a frame containing a MACPayload greater than the specified maximum length M over the data rate used to transmit the frame. Any frame received by an end-device or a network server containing a MACPayload greater than the specified maximum length M over the data rate used to receive the frame SHALL be silently discarded.

Table 2 – PHYPayload format

Size (octets)	1	7.. M	4
PHYPayload	MHDR	MACPayload	MIC

8.2 MAC header (MHDR field)

The MAC header specifies the frame type (FType) and the major version (Major) of the frame format of this Recommendation according to which the frame has been encoded (see Table 3).

Table 3 – MHDR format

Bits	[7:5]	[4:2]	[1:0]
MHDR	FType	RFU	Major

8.2.1 Frame types (FType bit field)

This Recommendation distinguishes among six different MAC frame types (see Table 4): Join-Request, Join-Accept, unconfirmed data uplink/downlink, and confirmed data uplink/downlink.

Table 4 – MAC frame types

FType	Description
000	Join-Request
001	Join-Accept
010	unconfirmed data uplink
011	unconfirmed data downlink
100	confirmed data uplink
101	confirmed data downlink
110	RFU
111	Proprietary

8.2.1.1 Join-Request and Join-Accept frames

Join-Request and Join-Accept frames are used by the over-the-air activation procedure described in clause 10.2.

8.2.1.2 Data frames

Data frames transfer MAC commands and application data, which can be combined into a single frame. A confirmed data frame SHALL be acknowledged by the receiver, whereas an unconfirmed data frame SHALL NOT be acknowledged. Proprietary frames MAY be used to implement non-standard formats that are not interoperable with standard frames but SHALL be used only among end-devices that have a common understanding of the proprietary extensions.

NOTE – A detailed timing diagram of the acknowledge mechanism is shown in Appendix I.

Frame integrity is ensured in different ways for different frame types and is described per frame type below.

8.2.2 Major data frame version (Major bit field)

The major version specifies the format of the frames (see Table 5) exchanged in the join procedure (see clause 10.2.4) and the first four octets of the MACPayload as described in clause 8. For each major version, end-devices MAY implement different minor versions of the frame format. The minor version used by an end-device SHALL be made known to the network server beforehand using out-of-band communication (e.g., as part of the end-device personalization information).

Table 5 – Major list

Major	Description
00	LoRaWAN R1
01...11	RFU

8.3 MAC payload of data frames (MACPayload)

The MAC payload of data frames contains a frame header (FHDR) followed by an OPTIONAL port field (FPort) and an OPTIONAL frame payload field (FRMPayload).

A frame with a valid FHDR, no FOpts (FOptsLen=0), no FPort and no FRMPayload is a valid frame.

8.3.1 Frame header (FHDR)

An FHDR contains the short end-device address (DevAddr), a frame control octet (FCtrl), a 2-octet frame counter (FCnt) and up to 15 octets of frame options (FOpts) to transport MAC commands (see Table 6).

Table 6 – FHDR format

Size (octets)	4	1	2	0..15
FHDR	DevAddr	FCtrl	FCnt	FOpts

For downlink frames, the FCtrl content of the frame header is given in Table 7.

Table 7 – FCtrl downlink frames format

Bits	7	6	5	4	[3..0]
FCtrl	ADR	RFU	ACK	FPending	FOptsLen

For uplink frames, the FCtrl content of the frame header is given in Table 8.

Table 8 – FCtrl uplink frame format

Bits	7	6	5	4	[3..0]
FCtrl	ADR	ADRACKReq	ACK	ClassB	FOptsLen

8.3.1.1 Adaptive data-rate control in frame header (ADR, ADRACKReq in FCtrl)

This Recommendation allows end-devices to use any of the possible data rates and transmission (TX) power individually. This feature is used by network servers to adapt and optimize the number of retransmissions, the data rate, and the TX power of end-devices. This is referred to as the adaptive data rate (ADR) and, when it is enabled, end-devices will be optimized to use the fastest data rate and minimum TX power possible.

ADR control may not be possible when radio channel attenuation changes rapidly and/or continuously. When the network is unable to control the data rate of an end-device, the end-device's application layer SHOULD control it. It IS RECOMMENDED that a variety of different data rates be used in this case. The application layer SHOULD always try to minimize the aggregated airtime, given the network conditions.

If the uplink ADR bit is set, the network may control the number of retransmissions, the data rate, and the TX power of the end-device through the appropriate MAC commands. If the ADR bit is unset, the network server SHALL accept that the end-device MAY not comply with any attempt to control the number of retransmissions, the data rate, or the TX power of the end-device regardless of the received signal quality. The network MAY still send commands to inform the end-device of the recommended configuration using the **LinkADRReq** command. An end-device SHOULD accept the channel mask controls present in **LinkADRReq**, even when the ADR bit is not set. The end-device SHALL respond to all **LinkADRReq** commands with a **LinkADRAns** indicating which command elements were accepted and which were rejected. This behaviour differs from when the uplink ADR bit is set, in which case the end-device accepts or rejects the entire command.

NOTE 1 – A network server may not infer any actual end-device state in the case where the uplink ADR bit is not set, regardless of the state of the individual Status bits of **LinkADRAns**. These are provided for offline debugging.

When the downlink ADR bit is set, it informs the end-device that the network server is able to send ADR commands. The end-device MAY set/unset the uplink ADR bit independently.

When the downlink ADR bit is unset, it signals the end-device that, owing to rapid changes of the radio channel, the network temporarily cannot estimate the best data rate. In that case, the end-device has the choice to:

- unset the ADR uplink bit and control its uplink data rate, TX power and channel plan following its own strategy. This SHOULD be the typical strategy for a mobile end-device; or

- ignore it (keep the uplink ADR bit set) and apply the normal ADR backoff algorithm in the absence of downlinks. This SHOULD be the typical strategy for a stationary end-device.

The ADR bit MAY be set and unset on demand by the end-device or a network server. However, whenever possible, the ADR scheme SHOULD be enabled to increase the battery life of the end-device and maximize the network capacity.

NOTE 2 – Even mobile end-devices are immobile most of the time. Depending on its state of mobility, an end-device can request that the network optimize its data rate using ADR.

Default TX power is the maximum transmit power allowed for an end-device, considering its capabilities and any applicable regional regulatory constraints. End-devices SHALL use this power level until the network attempts to change it using the **LinkADRReq** MAC command, or if the end-device has unset the ADR bit.

The default data rate is the minimum data rate allowed for an end-device, considering its capabilities and any applicable regional regulatory constraints. An end-device using activation by personalization (ABP, see clause 10), with the ADR bit set, SHALL use this data rate until the network requests a higher data rate through the **LinkADRReq** MAC command. End-devices which use OTAA follow a join procedure (see clause 10) which determines the initial uplink data rate.

If an end-device wishes to check for connectivity loss or if an end-device whose data rate is optimized by the network uses a data rate higher than its default data rate or a TX power lower than its default, the end-device SHALL periodically validate whether the network is still receiving the uplink frames. Each time the uplink frame counter is incremented (for each new uplink frame, because repeated transmissions do not increment the frame counter), the end-device SHALL increment an ADRACKCnt counter. After ADR_ACK_LIMIT uplinks ($\text{ADRACKCnt} \geq \text{ADR_ACK_LIMIT}$) without receiving a Class A downlink response, the end-device SHALL set the ADR acknowledgment request bit (ADRACKReq) on uplink transmissions. The network shall respond with a Class A downlink frame within the next ADR_ACK_DELAY frames. A Class A downlink frame received following an uplink frame SHALL reset the ADRACKCnt counter. Upon receipt of any Class A downlink, the end-device SHALL clear the ADRACKReq bit. The downlink ACK bit does not need to be set because any Class A downlink frame received by the end-device indicates that the network has received uplinks from this end-device. If no Class A downlink frame is received within the next ADR_ACK_DELAY uplinks (i.e., after a total of $\text{ADR_ACK_LIMIT} + \text{ADR_ACK_DELAY}$ transmitted frames), the end-device SHALL try to regain connectivity by first setting the TX power to the default power, then switching to the next lower data rate that provides a longer radio range. The end-device SHALL further lower its data rate step by step every time ADR_ACK_DELAY uplink frames are transmitted. Once the end-device has reached the default data rate, and transmitted for ADR_ACK_DELAY uplinks with ADRACKReq=1 without receiving a downlink, it SHALL re-enable all default uplink frequency channels and reset NbTrans to its default value of 1. Furthermore, if at any point during the backoff the resulting configuration results in an invalid combination of TX power, data rate or channel mask, the end-device SHALL immediately re-enable all default channels and use the maximum TX power permissible for and available to this end-device.

NOTE 3 – Other configurations of the end-device by the network are not modified during ADR backoff. Specifically, configurations that affect downlink connectivity (controlled by **RXParamsSetupReq**, **DLChannelReq**, **RXTimingSetupReq**, and **TXParamSetupReq**) are not modified during ADR backoff.

For fixed channel plan regions, end-devices SHALL enable all channels. For dynamic channel plan regions, end-devices SHALL enable the region's default channels and make no change to the configuration of the dynamically configured channels.

NOTE 4 – Not requiring an immediate response to an ADR acknowledgment request provides flexibility to the network to schedule its downlinks in an optimal manner.

Table 9 provides an example of a data rate backoff sequence, assuming ADR_ACK_LIMIT is set to 64 and ADR_ACK_DELAY is equal to 32.

Table 9 – Example of a data rate backoff sequence

ADRACKCnt	ADRACKReq	Data rate	TX power	NbTrans	Channel mask
0 to 63	0	DR1	Max –9 dBm	3	Normal operations channel mask
64 to 95	1	No change	No change	No change	No change
96 to 127	1	No change	Default	No change	No change
128 to 159	1	DR0 (Default)	Default	No change	No change
≥ 160	1	DR0 (Default)	Default	1	For dynamic channel plans: re-enable default channels. For fixed channel plans: All channels enabled

8.3.1.2 Frame acknowledge bit and acknowledgment procedure (ACK in FCtrl1)

When receiving a confirmed data frame, the receiver responds with a data frame that has the acknowledgment bit (ACK) set. If the network receives such a confirmed frame, it SHOULD send an acknowledgment. If the network sends an acknowledgement, it SHALL send it using one of the Class A receive windows opened by the end-device after the send operation. If an end-device receives such a confirmed frame in one of its Class A receive windows, it SHALL transmit an acknowledgment with its next uplink. If an end-device receives such a confirmed frame outside of its Class A receive windows, i.e., in a Class B ping slot (see clause 13) or in RXC (see clause 19), it SHOULD send an acknowledgement.

Acknowledgments SHALL only be sent in response to the latest frame received and SHALL NOT be transmitted more than NbTrans times. The network SHALL only send an acknowledgment in the Class A receive windows (RX1/RX2) of the confirmed uplink that requested it.

NOTE – To allow an end-device to be as simple as possible and have as few states as possible, it may transmit an explicit (possibly empty) acknowledgment data frame immediately after receiving a data frame requiring a confirmation. Alternatively, an end-device may defer the transmission of an acknowledgment to piggyback it with its next data frame.

8.3.1.3 Retransmission procedure

Downlink frames

A downlink confirmed or unconfirmed frame SHALL NOT be retransmitted using the same frame counter value. In the case of a confirmed downlink, if the acknowledgement is not received, the application server is notified and may decide to transmit a new confirmed frame.

Uplink frames

Uplink confirmed and unconfirmed frames are transmitted NbTrans times (see clause 9.2) unless a valid Class A downlink is received following one of the transmissions. The NbTrans parameter can be used by a network server to control the redundancy of end-device uplinks to achieve a given quality of service. An end-device SHALL perform frequency hopping as usual between repeated transmissions. It SHALL wait after each repetition until the receive windows have expired. The delay

between retransmissions is at the discretion of the end-device and MAY be different for each end-device. When an end-device has requested an ACK from the network but has not yet received it, it SHALL wait RETRANSMIT_TIMEOUT seconds after the RECEIVE_DELAY2 seconds have elapsed, after the end of the previous uplink transmission before sending a new uplink (repetition or new frame). The RETRANSMIT_TIMEOUT delay is not required between unconfirmed uplinks, or after the ACK has been successfully demodulated by the end-device. An example of the RETRANSMIT_TIMEOUT value may be found in [b-LoRaWAN RP002]. The retransmission backoff mechanism, defined in clause 11, may also extend the interval between retransmissions, if applicable.

End-devices SHALL stop any further retransmission of an uplink confirmed frame if a corresponding downlink acknowledgment frame is received.

End-devices SHALL also stop any further retransmission of an uplink unconfirmed frame whenever a valid downlink frame is received in a Class A receive window.

If the network receives more than NbTrans transmissions of the same uplink frame having the ADR bit set, this may indicate a replay attack or a malfunctioning end-device, and therefore the network SHALL silently discard the extra frames.

NOTE – A network server that detects a replay attack may take additional measures, such as reducing the NbTrans parameter to 1 or discarding uplink frames received over a channel that was already used by an earlier transmission of the same frame, or by some other unspecified mechanism.

8.3.1.4 Frame pending bit (FPending in FCtrl, downlink only)

The frame pending bit (FPending) is used only in downlink communication. For Class A end-devices, FPending indicates that the network server has more data pending to be sent and therefore the end-device MAY send an uplink frame as soon as possible. For Class B end-devices, FPending indicates the priority in which conflicting ping slots of the end-device SHALL listen to in case of a collision (see clause 13.2).

An example use of the FPending bit is described in clause I.4.

8.3.1.5 Frame counter (FCnt)

There are two frame counters for each end-device. FCntUp is incremented by an end-device when a data frame is transmitted to a network server (uplink). FCntDown is incremented by a network server when a data frame is transmitted to an end-device (downlink). The network server tracks the uplink frame counter and generates the downlink counter for each end-device.

Whenever an OTAA end-device successfully processes a Join-Accept frame, the frame counters on the end-device (FCntUp) and the network side (FCntDown) are reset to 0.

For ABP end-devices, the frame counters are initialized to 0 by the manufacturer. ABP end-devices SHALL NOT reset the frame counters during the end-device's lifetime. If the end-device is susceptible to losing power during its lifetime (battery replacement, for example), the frame counters SHALL persist during such an event.

Subsequently, FCntUp is incremented with each uplink and FCntDown is incremented with each downlink. At the receiver side, the corresponding counter is kept in sync with the received value, provided the received value has been incremented compared to the current counter value, and the frame MIC field matches the MIC value computed locally using the appropriate network session key. FCntUp SHALL NOT be incremented in the case of multiple transmissions of a confirmed or unconfirmed frame (see NbTrans parameter). Network servers SHALL drop the application payload of the retransmitted frames and only forward a single instance to the appropriate application server.

A first uplink with FCntUp=0 sent by an ABP or an OTAA end-device after a successful join procedure SHALL be accepted by a network server, provided the MIC field is valid. Analogously, a

first downlink with `FCntDown=0` sent by a network server to an ABP or an OTAA end-device after a successful join procedure SHALL be accepted by the end-device, provided the MIC field is valid.

Frame counters are 32 bits wide. The `FCnt` field SHALL correspond to the least-significant 16 bits of the 32-bit frame counter (i.e., `FCntUp` for data frames sent uplink and `FCntDown` for data frames sent downlink).

The end-device SHALL NOT reuse the same `FCntUp` value with the same application or network session keys, except for retransmission.

The end-device SHALL NOT process any retransmission of the same downlink frame. Subsequent retransmissions SHALL be ignored without being processed.

NOTE 1 – This means that an end-device will only acknowledge receipt of a downlink confirmed frame `NbTrans` times. Similarly, an end-device will only generate `NbTrans` uplinks following receipt of a frame with the `FPending` bit set before incrementing its `FCntUp`.

NOTE 2 – As the `FCnt` field carries only the least-significant 16 bits of the 32-bit frame counter, the server must infer the 16 most-significant bits of the frame counter by observing the traffic.

8.3.1.6 Frame options (`FOptsLen` in `FCtrl`, `FOpts`)

The frame-options length field (`FOptsLen`) in `FCtrl` denotes the actual length of the frame options field (`FOpts`) included in the frame.

`FOpts` transports MAC commands of a maximum length of 15 octets that are piggybacked onto data frames; see clause 9 for a list of valid MAC commands.

If `FOptsLen=0`, the `FOpts` field SHALL be absent. If `FOptsLen≠0`, i.e., if MAC commands are present in the `FOpts` field, the `FPort` value 0 SHALL NOT be used (`FPort` SHALL either not be present or not equal to 0).

MAC commands SHALL NOT be present in the payload field and the frame options field simultaneously. Should this occur, the end-device SHALL silently discard the frame.

8.3.1.7 Class B enabled bit (`ClassB` in `FCtrl`, uplink only)

The `ClassB` bit set to 1, in an uplink, signals to the network server that the end-device has enabled class B and is now ready to receive scheduled downlink pings. Please refer to the Class B section of this Recommendation for the Class B specification.

8.3.2 Port field (`FPort`)

If the frame payload field is not empty, the port field SHALL be present (see Table 10).

If present, an `FPort` value of 0 indicates that the `FRMPayload` contains only MAC commands (see clause 9 for a list of valid MAC commands).

`FPort` values 1..223 (0x01..0xDF) are application-specific.

`FPort` value 224 is dedicated to the MAC layer test protocol.

NOTE – The purpose of the `FPort` value 224 is to run MAC compliance test scenarios over-the-air on final versions of end-devices, without having to rely on specific test versions of end-devices for practical aspects. The test protocol running at the application layer is defined in [b-LoRaWAN TS009].

`FPort` values 225..255 (0xE0..0xFF) are reserved for future use.

Table 10 – MACPayload format

Size (octets)	7..22	0..1	0..N
MACPayload	FHDR	FPort	FRMPayload

In Table 10, N is the number of octets of the application payload and SHALL be equal to or less than $N \leq M - 1 - (\text{length of FHDR in octets})$, where M is the maximum MACPayload length. The valid ranges of both N and M depend on the physical layer and the regional channel plan.

8.3.3 MAC frame payload encryption (FRMPayload)

If a data frame carries a payload (FRMPayload), it SHALL be encrypted before the message integrity code (MIC) is calculated.

The encryption scheme is based on the generic algorithm described in [IEEE 802.15.4] Annex B, using advanced encryption standard (AES) encryption with a key length of 128 bits. AES encryption is defined in [NIST FIPS 197].

Key K depends on the FPort of the data frame (see Table 11).

Table 11 – FPort list

FPort	K
0	NwkSKey
1..255	AppSKey

The encrypted fields are $pld = \text{FRMPayload}$.

For each data frame, the algorithm defines a sequence of blocks A_i as specified in Table 12, for $i = 1..k$ where:

- $k = \text{ceil}(\text{len}(pld) / 16)$,
- "ceil" designates rounding to the immediately greater integer value, and
- "len" is the number of bytes.

Table 12 – A_i format

Size (octets)	1	4	1	4	4	1	1
A_i	0x01	$4 \times 0x00$	Dir	DevAddr	FCntUp or FCntDown	0x00	i

The direction field (Dir) is 0 for uplink frames and 1 for downlink frames.

The blocks A_i SHALL be encrypted to obtain a sequence S of blocks S_i as follows:

$$S_i = \text{aes128_encrypt}(K, A_i) \text{ for } i = 1..k$$

$$S = S_1 | S_2 | .. | S_k$$

Encryption and decryption of the payload SHALL be calculated as follows:

$$\text{FRMPayloadPad} = (pld | \text{pad}_{16}) \text{ xor } S$$

$$\text{FRMPayload} = \text{FRMPayloadPad}[0..\text{len}(pld) - 1]$$

8.4 Message integrity code (MIC)

The MIC is calculated over all the fields in the frame.

$$msg = \text{MHDR} | \text{FHDR} | \text{FPort} | \text{FRMPayload}$$

where $\text{len}(msg)$ denotes the length of the frame in octets.

The MIC SHALL be calculated as follows, where CMAC is the cipher-based message authentication code (see [IETF RFC 4493]):

$$\text{CMAC} = \text{aes128_cmac}(\text{NwkSKey}, B_0 | msg)$$

$$\text{MIC} = \text{CMAC}[0..3]$$

where block B_0 is defined in Table 13.