ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 71168— 2023

Информационные технологии

ИНТЕРНЕТ ВЕЩЕЙ

Спецификация LoRaWAN RU

Издание официальное

Москва
Российский институт стандартизации
2024

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Ассоциацией участников рынка интернета вещей

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 194 «Кибер-физические системы»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 22 декабря 2023 г. № 1617-ст

4 ВВЕДЕН ВПЕРВЫЕ

5 ДЕЙСТВУЕТ ВЗАМЕН ПНСТ 516—2021

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Оформление. ФГБУ «Институт стандартизации», 2024

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 2
3 Термины и определения 2
4 Сокращения и обозначения 2
5 Классы устройств LoRaWAN RU 3
6 Оконечные устройства класса A 4
6.1 Режимы связи и окна приема в оконечных устройствах класса A 4
6.2 Формат сообщений на MAC-уровне 5
6.3 MAC-команды 14
6.4 Активация оконечного устройства 31
6.5 Задержка повторных передач 42
7 Оконечные устройства класса C 42
7.1 Режим связи оконечного устройства 42
7.2 MAC-команды 43
8 Примеры реализации 44
8.1 Диаграмма передачи восходящего сообщения с подтверждением 44
8.2 Диаграмма передачи нисходящего сообщения с подтверждением 45
8.3 Диаграмма передачи очереди нисходящих сообщений 46
9 Региональные параметры 47
9.1 RU864-870 47
Библиография 53

Введение

Настоящий стандарт определяет телеметрический протокол с адаптивной полосой (LoRaWAN RU), оптимизированный для оконечных устройств с батарейным питанием, которые могут быть мобильными или стационарными.

В базовой конфигурации сеть LoRaWAN RU имеет топологию «звезда из звезд» (star-of-stars). В расширенной конфигурации сеть LoRaWAN RU может иметь конфигурацию, заложенную разработчиком в оконечное устройство (P-2-P, Mesh-сети и др.). В базовой конфигурации шлюзы* ретранслируют сообщения между оконечными устройствами** и центральным сервером сети. Сервер сети маршрутизирует пакеты с каждого устройства сети на соответствующий сервер приложений.

Шлюзы подключаются к серверу сети через IP-соединения. Оконечные устройства используют прямую передачу данных на один или несколько шлюзов с линейно-частотной модуляцией***. Обмен данными двусторонний, однако предполагается, что объем данных в восходящей линии связи (от оконечного устройства к серверу сети) будет преобладать над объемом данных в нисходящей линии связи (от сервера сети к оконечным устройствам).

Связь между оконечными устройствами и шлюзами распределяется по различным частотным каналам и скоростям передачи данных. Выбор скорости передачи данных — это компромисс между максимальной дальностью связи и минимальной продолжительностью сообщения в радиоэфире. Одновременная связь с разными оконечными устройствами на одной частоте, но с разными скоростями передачи данных не создает взаимных помех. Скорость передачи данных по протоколу LoRaWAN RU составляет от 0,3 до 50 кбит/с. Чтобы максимально увеличить время автономной работы оконечных устройств и общую пропускную способность сети, сервер сети может управлять скоростью передачи данных и выходной мощностью радиопередатчика для каждого оконечного устройства индивидуально, посредством схемы адаптивной скорости передачи данных (ADR, adaptive data rate).

Оконечные устройства могут передавать данные по любому доступному частотному каналу в любое время, используя любую доступную скорость передачи данных, если соблюдаются следующие правила:

- оконечное устройство для каждой передачи выбирает частотный канал псевдослучайным образом. Данное распределение по частотам делает систему связи более устойчивой к помехам;
- оконечное устройство учитывает ограничение на максимальный рабочий цикл передачи (DutyCycle) для используемого диапазона частот;
- оконечное устройство учитывает ограничение на максимальную длительность передачи (TimeOnAir) для используемого диапазона частот.

Примечание — Максимальный рабочий цикл передачи и максимальная длительность передачи на каждый поддиапазон являются региональными параметрами и определены в 6.1.

В радиоканале используется ЛЧМ-модуляция****, адаптированная для устройств с низким энергопотреблением и низкой скоростью передачи. В настоящем стандарте устройства, реализующие функциональность в объеме большем, чем для класса А, называются «оконечными устройствами высшего класса».

---

* Шлюзы также известны как концентраторы или базовые станции.

** Оконечные устройства также называют «мотами».

*** В настоящем стандарте не рассматривается поддержка промежуточных элементов — повторителей и рассматриваются ограничения полезной нагрузки для издержек инкапсуляции. Повторитель определен как использующий протокол LoRaWAN RU в качестве механизма обратного рейса.

**** Линейная частотная модуляция.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

ИНТЕРНЕТ ВЕЩЕЙ

Спецификация LoRaWAN RU

Information technology. Internet of things. LoRaWAN RU specification

Дата введения — 2024—07—01

1 Область применения

Целью настоящего стандарта является создание основы для скоординированной разработки инфраструктуры в области интернета вещей (IoT, Internet of Things).

Областью применения описанной в настоящем стандарте технологии связи является обмен данными с бытовым и промышленным оборудованием (приборы, датчики, пломбы, исполнительные устройства и т. п.), имеющим свойства:

- низкие требования к пропускной способности каналов связи;
- высокие требования к помехоустойчивости каналов связи;
- высокие требования к времени автономной работы (до 10 лет и более);
- предназначены для хранения и/или передачи общедоступной информации и информации ограниченного доступа, не содержащей сведений, относимых к государственной тайне (конфиденциальная информация);
- не предназначены для хранения и/или передачи информации, содержащей сведения, составляющие государственную тайну.

Настоящий стандарт беспроводной связи обеспечивает обратную совместимость со стандартами LoRaWAN v.1.1 final release 11.11.2017 (кроме устройств класса «B») и LoRaWAN v.1.0.2 final 07.2016 (кроме устройств класса «B»).

Выбор конкретных алгоритмов шифрования данных определяет разработчик оконечного устройства в зависимости от целевого назначения оконечного устройства и действующих нормативных документов. Например, для систем класса критической информационной инфраструктуры (КИИ) могут использоваться криптографические преобразования. Используемые в оконечном устройстве алгоритмы шифрования данных должны быть указаны в эксплуатационной документации на соответствующее оконечное устройство. Рекомендуется использовать алгоритм блочного шифрования. Описание и выбор конкретных алгоритмов шифрования данных выходят за рамки настоящего стандарта.

В настоящем стандарте используются следующие форматы записей:

- MAC-команды записываются в формате *LinkCheckReq*;
- константы записываются в формате RECEIVE_DELAY1.

В настоящем стандарте структура сообщений изложена с учетом:

- порядка следования байтов и битов для всех полей — «от младшего к старшему» (little-endian);
- бита RFU (Reserved For Future), обозначающего поле для будущего использования. По умолчанию данный бит должен быть установлен на нуль передатчиком сообщения и проигнорирован на приемной стороне.

Издание официальное

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий стандарт:

ГОСТ Р ИСО/МЭК 7498-1 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применены термины по ГОСТ Р ИСО/МЭК 7498-1, а также следующие термины с соответствующими определениями:

3.1 **активация** (activation): Процедура присоединения оконечного устройства к сети.

3.2 **активация «по воздуху»;** OTAA (over the air activation, OTAA): Способ активации оконечного устройства в сети через запрос-ответ.

3.3 **активация через персонализацию;** ABP (activation by personalization, ABP): Способ активации оконечного устройства в сети через предустановленные параметры контекста сеанса связи.

3.4 **контекст сеанса связи** (session context): Набор параметров сетевого сеанса и параметров сеанса приложения.

3.5 **многоадресная рассылка** (multicast): Режим передачи нисходящего сообщения нескольким оконечным устройствам одновременно.

3.6 **оконечное устройство** (end-device): Программно-аппаратное решение, являющееся узлом сети, который может обмениваться сообщениями по протоколу LoRaWAN RU.

3.7 **параметры сеанса приложения** (application session parameters): Набор ключей и счетчиков, поддерживаемый оконечным устройством и сервером приложений.

3.8 **параметры сетевого сеанса** (network session parameters): Набор ключей и счетчиков, поддерживаемый оконечным устройством и сервером сети.

3.9 **рабочий цикл устройства** (duty-cycle): Безразмерная величина, обратная скважности и равная отношению длительности передаваемого радиосообщения к периоду передачи.

3.10 **сеансовый ключ** (session key): Производный ключ от первичных ключей, хранящихся в оконечном устройстве.

3.11 **uplink-канал** (uplink): Канал связи восходящих сообщений от оконечного устройства к серверу сети.

3.12 **downlink-канал** (downlink): Канал связи нисходящих сообщений от сервера сети к оконечному устройству.

4 Сокращения и обозначения

В настоящем стандарте применены следующие сокращения и обозначения:

ОЗУ	—	оперативное запоминающее устройство;
MIC	—	код целостности сообщения;
AppKey	—	прикладной первичный ключ оконечного устройства;
AppSKey	—	сеансовый ключ приложения — для кодирования прикладных данных в нисходящих и восходящих сообщениях;
DevAddr	—	короткий адрес оконечного устройства;

DevEUI	—	идентификатор оконечного устройства;
DevNonce	—	идентификатор запросов на присоединение к сети;
FNwkSIntKey	—	сетевой сеансовый ключ целостности восходящих сообщений;
JoinEUI	—	идентификатор сервера присоединения к сети;
Join-Accept	—	подтверждение присоединения (переприсоединения) к сети;
Join-Request	—	запрос на присоединение к сети;
JoinNonce	—	идентификатор подтверждений присоединения (переприсоединения) к сети;
JSEncKey	—	ключ кодирования Join-Accept в случае получения запроса Rejoin-Request;
JSIntKey	—	ключ целостности восходящего сообщения с запросом Rejoin-Request первого типа и нисходящего сообщения ответа Join-Accept;
NwkKey	—	сеансовый первичный ключ оконечного устройства;
NwkSEncKey	—	сетевой сеансовый ключ кодирования MAC-команд в нисходящих и восходящих сообщениях;
Rejoin-Request	—	запрос на повторное присоединение к сети;
RFU	—	зарезервировано для будущих применений;
SNwkSIntKey	—	сетевой сеансовый ключ целостности нисходящих сообщений;
«||»	—	знак конкатенации строк;
«[x:y]»	—	последовательность битов, где «x» — старший бит, а «y» — младший бит;
«Pad_{16}»	—	функция добавления нулевых байтов таким образом, чтобы общая длина данных стала кратной 16.

5 Классы устройств LoRaWAN RU

Все оконечные устройства должны реализовать базовую функциональность класса A, определенного в настоящем стандарте.

Некоторые оконечные устройства могут дополнительно реализовать вариант протокола с изменением до класса C, определенного в настоящем стандарте. Любое оконечное устройство должно поддерживать совместимость с классом A.

Сеть LoRaWAN RU различает базовый класс устройств (класс A) и класс с дополнительными функциями (класс C) (см. рисунок 1).

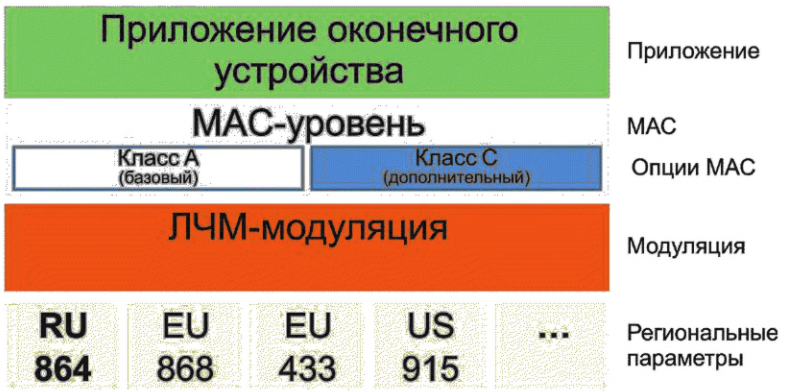

Рисунок 1 — Классы устройств LoRaWAN RU

Двунаправленные оконечные устройства (класс A): Оконечные устройства класса A предназначены для обеспечения двунаправленной связи, в которой за каждой передачей сообщения оконечным устройством в восходящую линию связи следуют два коротких окна приема для нисходящей линии связи. Интенсивность передачи каждым оконечным устройством зависит от его собственной потреб-

ности в связи с небольшим отклонением времени начала передачи сообщения на случайную величину (протокол ALOHA). Класс А обеспечивает экономичный расход энергии и подходит в случаях, когда приемлема связь по нисходящей линии связи от сервера только после передачи по восходящей линии связи. Сервер сети для передачи сообщения по нисходящей линии связи должен дождаться следующего сообщения от оконечного устройства по восходящей линии связи.

Двунаправленные оконечные устройства с максимальными окнами приема (класс С): У оконечных устройств класса С большую часть времени открыто одно из окон приема (за исключением времени передачи сообщения по восходящей линии связи). Оконечное устройство класса С будет расходовать больше энергии по сравнению с устройством класса А, но при этом иметь минимальную задержку передачи сообщения от сервера сети к оконечному устройству.

6 Оконечные устройства класса А

6.1 Режимы связи и окна приема в оконечных устройствах класса А

6.1.1 Режимы связи оконечного устройства

Оконечное устройство различает два режима связи:

- передача восходящего сообщения к серверу сети;
- прием нисходящего сообщения от сервера сети.

Настоящий протокол предполагает наличие между оконечным устройством и сервером сети одного нисходящего радиоканала и множества восходящих радиоканалов. С целью повышения емкости нисходящего радиоканала в структуре нисходящего сообщения отсутствует поле «Циклическая контрольная сумма» (CRC).

6.1.2 Окна приема (класс А)

После каждой передачи восходящего сообщения оконечное устройство должно открыть два коротких окна приема. Момент открытия окон RX1 и RX2 приема нисходящих сообщений задается относительно времени окончания передачи восходящего сообщения (рисунок 2).

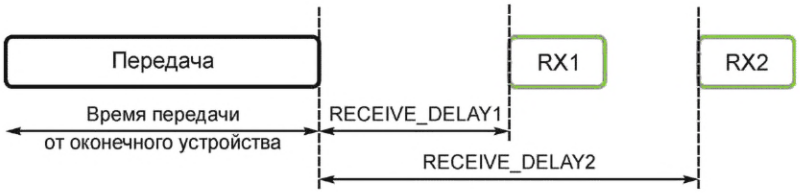

Рисунок 2 — Временные сегменты приема оконечного устройства

6.1.2.1 Канал, скорость передачи данных и время открытия первого окна приема

Первое окно приема RX1 использует частоту, значение которой зависит от частоты в восходящей линии связи, и скорость приема данных в окне RX1, которая зависит от скорости передачи данных предыдущего восходящего сообщения. Окно приема RX1 открывается с задержкой RECEIVE_DELAY1 секунд (+/− 20 микросекунд) после окончания модуляции восходящего сообщения (RECEIVE_DELAY1 и RECEIVE_DELAY2 определены в 9.1.8). Отношение между скоростью передачи данных в восходящей линии связи и скоростью передачи нисходящих данных в окне приема RX1 является региональным параметром и определено в 9.1.7. По умолчанию скорость передачи нисходящего сообщения в первом окне приема RX1 равна скорости передачи данных, использованной при передаче последнего восходящего сообщения.

6.1.2.2 Канал, скорость передачи данных и время открытия второго окна приема

Второе окно приема RX2 использует фиксированно заданные частоту приема и скорость передачи данных и открывается с задержкой RECEIVE_DELAY2 секунд (+/− 20 микросекунд) после окончания модуляции восходящего сообщения (RECEIVE_DELAY1 и RECEIVE_DELAY2 определены в 9.1.8). Частота и скорость передачи данных, используемые в окне RX2, могут быть изменены с помощью MAC-команд (см. 6.3). Значения частоты и скорости передачи данных по умолчанию являются региональными параметрами и указаны в 9.1.8.

6.1.2.3 Длительность окон приема

Длина окна приема должна быть не менее времени, необходимого радиоприемнику оконечного устройства для эффективного распознавания преамбулы нисходящего сообщения.

6.1.2.4 Активность приемника во время окна приема

Если поле «Преамбула» (Preamble) обнаружено во время одного из окон приема, то радиоприемник остается активным до окончания демодуляции нисходящего сообщения. Если сообщение распознано и затем демодулировано в пределах первого окна приема RX1 и кадр был адресован этому оконечному устройству после проверки адреса и MIC, то оконечное устройство не должно открывать второе окно приема RX2.

6.1.2.5 Отправка сообщения на оконечное устройство

Если необходимо передать оконечному устройству нисходящее сообщение, то сервер сети должен инициировать передачу точно в начале хотя бы одного из двух окон приема. Если нисходящее сообщение передается в течение обоих окон, то должны передаваться идентичные пакеты в каждое окно.

6.1.2.6 Примечание об окнах приема

Оконечное устройство не должно передавать следующее восходящее сообщение до того, как получит нисходящее сообщение в первом или втором окне приема, открытом по факту предыдущей передачи, или пока не закроется второе окно приема, открытое по факту предыдущей передачи.

6.1.2.7 Прием или передача по другим протоколам

Оконечное устройство может принимать или передавать данные по другим протоколам в свободные промежутки времени между передачей по протоколу LoRaWAN RU и окнами приема RX1/RX2 до тех пор, пока устройство не нарушит региональных регуляторных ограничений и требований протокола LoRaWAN RU.

6.2 Формат сообщений на MAC-уровне

Структура восходящего радиосообщения на физическом уровне приведена на рисунке 3.

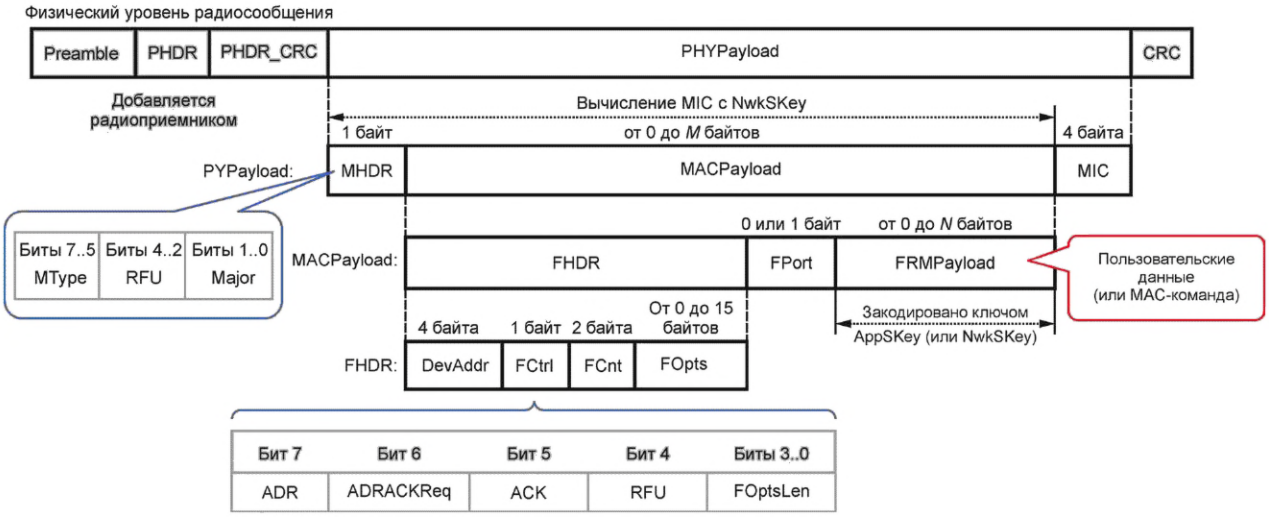

Рисунок 3 — Структура восходящего радиосообщения на физическом уровне

Структура нисходящего радиосообщения на физическом уровне приведена на рисунке 4.

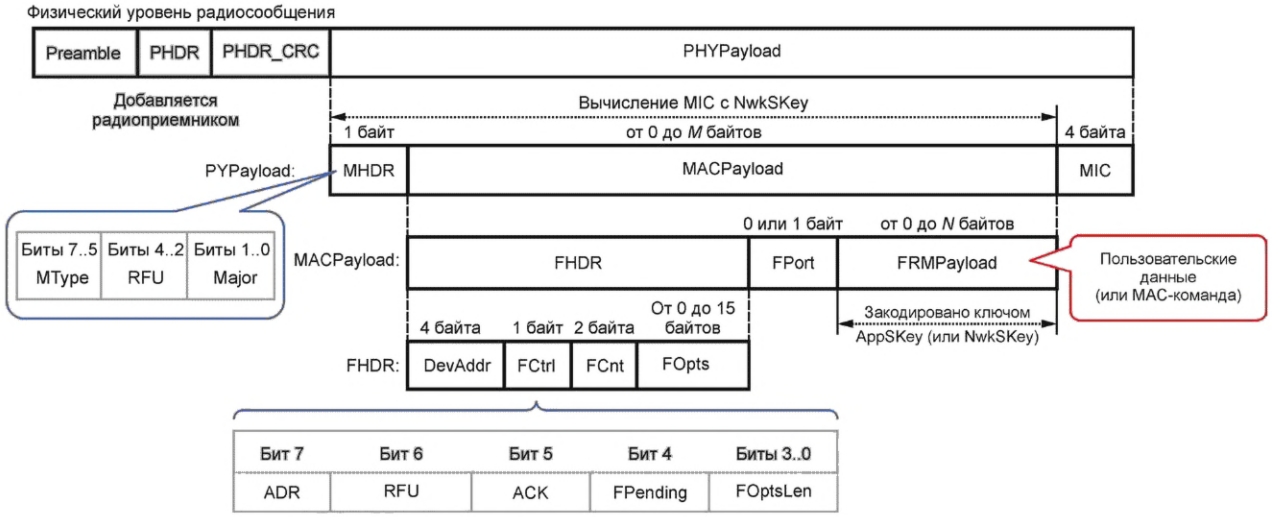

Рисунок 4 — Структура нисходящего радиосообщения на физическом уровне

Структура восходящего радиосообщения с запросом на присоединение к сети приведена на рисунке 5.



Рисунок 5 — Структура восходящего радиосообщения с запросом на присоединение к сети

Структура нисходящего радиосообщения с подтверждением о присоединении к сети приведена на рисунке 6.

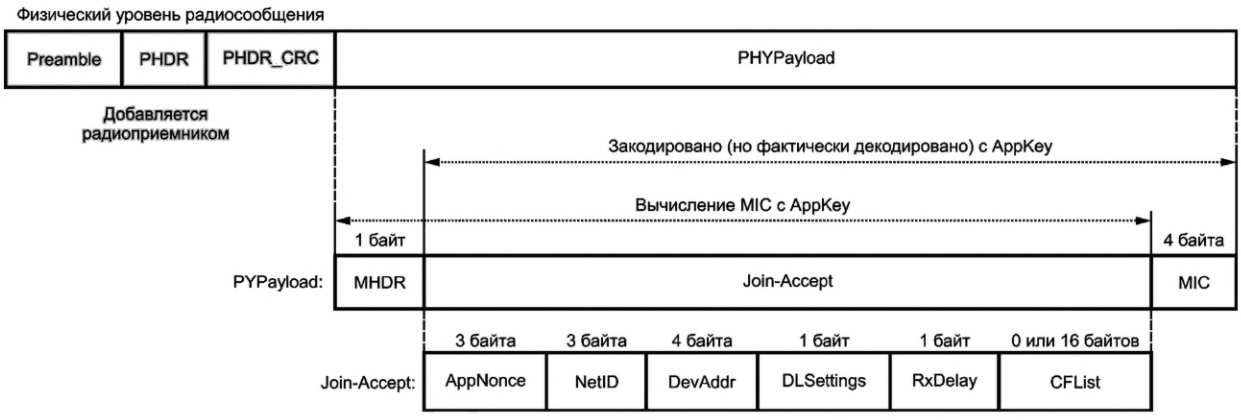

Рисунок 6 — Структура нисходящего радиосообщения с подтверждением о присоединении к сети

Поле «Преамбула» (Preamble) имеет длину 8 символов ЛЧМ и предназначена для синхронизации приемного устройства перед демодулированием радиосообщения.

В радиосообщениях используется режим явного заголовка радиопакета, в который включены поле «Физический заголовок» (PHDR) и поле «Контрольная сумма физического заголовка» (PHDR_CRC).

В поле «Физический заголовок» (PHDR) передается: длина поля «Данные» (PHYPayload) в байтах, масштаб кода исправления ошибок и наличие/отсутствие поля «Циклическая контрольная сумма» (CRC).

Поле «Циклическая контрольная сумма» (CRC) имеет длину 2 байта и предназначена для контроля целостности поля «Данные» (PHYPayload).

Поле «Данные» (PHYPayload) имеет один из следующих форматов:

- в восходящем и нисходящем сообщениях поле «Данные» (PHYPayload) начинается с поля «MAC-заголовок» (MHDR) размером 1 байт, затем передается поле «MAC-сообщение» (MACPayload), и заканчивается полем «Код целостности сообщения» (MIC) размером 4 байта.

Поле «MAC-заголовок» (MHDR) описано в 6.2.2.

Поле «MAC-сообщение» (MACPayload) описано в 6.2.3;

- в сообщении с запросом на присоединение к сети поле «Данные» (PHYPayload) начинается с поля «MAC-заголовок» (MHDR) размером 1 байт, затем передается поле «Запрос на присоединение к сети» (Join-Request) или «Запрос на повторное присоединение к сети» (Rejoin-Request), и завершается полем «Код целостности сообщения» (MIC) размером 4 байта. Поле «Запрос на присоединение к сети» (Join-Request) описано в 6.4.2.2. Поле «Запрос на повторное присоединение к сети» (Rejoin-Request) описано в 6.4.2.4;

- в сообщении с подтверждением о присоединении к сети поле «Данные» (PHYPayload) начинается с поля «MAC-заголовок» (MHDR) размером 1 байт, затем передается поле «Подтверждение присоединения (переприсоединения) к сети» (Join-Accept). При передаче «Подтверждение присоединения (переприсоединения) к сети» (Join-Accept) поле «Код целостности сообщения» (MIC) включено в состав передаваемых данных и не передается отдельным полем. Поле «Подтверждение присоединения (переприсоединения) к сети» (Join-Accept) описано в 6.4.2.3.

6.2.1 Поле «Данные» (PHYPayload)

Структура поля «Данные» (PHYPayload) приведена на рисунке 7.

Размер (в байтах)	1	*7…M*	4
Данные (PHYPayload)	MAC-заголовок (MHDR)	MAC-сообщение (MACPayload)	Код целостности сообщения (MIC)

Рисунок 7 — Структура поля «Данные» (PHYPayload)

Максимальная длина (*M*) поля «MAC-сообщение» (MACPayload) зависит от скорости передачи сообщения и является региональным параметром (см. 9.1.6). При передаче подтверждения присоединения к сети длина MAC-сообщения (MACPayload) равна длине поля «Подтверждение присоединения (переприсоединения) к сети» (Join-Accept) (см. 6.4.2.3) и поле «Код целостности сообщения» (MIC) отсутствует.

6.2.2 Поле «MAC-заголовок» (MHDR)

Структура поля «MAC-заголовок» (MHDR) с указанием битов полей приведена на рисунке 8.

Биты	7..5	4..2	1..0
MAC-заголовок (MHDR)	Тип сообщения (MType)	RFU	Основная версия формата данных (Major)

Рисунок 8 — Структура поля «MAC-заголовок» (MHDR)

6.2.2.1 Поле «Тип сообщения» (MType)

Согласно настоящему стандарту в протоколе LoRaWAN RU различают восемь различных типов MAC-сообщений (см. таблицу 1).

Таблица 1 — Типы MAC-сообщений

Значение поля	Описание
000	Запрос на присоединение к сети (Join-Request)
001	Подтверждение присоединения (переприсоединения) к сети (Join-Accept)
010	Неподтверждаемые восходящие данные (Unconfirmed Data Up)
011	Неподтверждаемые нисходящие данные (Unconfirmed Data Down)
100	Подтверждаемые восходящие данные (Confirmed Data Up)
101	Подтверждаемые нисходящие данные (Confirmed Data Down)
110	Запрос на повторное присоединение к сети (Rejoin-Request)
111	Сообщения собственного протокола (Proprietary protocol messages)

а) Сообщения «Запрос на присоединение к сети» (Join-Request), «Подтверждение присоединения (переприсоединения) к сети» (Join-Accept) и «Запрос на повторное присоединение к сети» (Rejoin-Request)

Сообщения: «Запрос на присоединение к сети» (Join-Request), «Подтверждение присоединения (переприсоединения) к сети» (Join-Accept) и «Запрос на повторное присоединение к сети» (Rejoin-Request) используются в процедуре активации по воздуху, описанной в 6.4.2, и в целях роуминга.

б) Сообщения с данными

Сообщения с данными используются для передачи MAC-команд и данных приложений (прикладных данных), которые могут быть объединены вместе в одном сообщении. Подтвержденное сообщение с данными, требующее уведомления о получении сообщения, должно быть подтверждено получателем, в то время как неподтвержденное сообщение не требует отправки уведомления (подробная временная диаграмма работы механизма подтверждения приведена в разделе 8). Собственные типы сообщений могут использоваться для реализации нестандартных форматов сообщений, которые не совместимы со стандартными сообщениями, но должны использоваться для поддержки устройств (между устройствами), имеющих общее понимание собственных (нестандартных) расширений. Когда оконечное устройство или сетевой сервер получают сообщение неизвестного нестандартного формата, они должны его проигнорировать.

Целостность сообщения обеспечивается разными способами для разных типов сообщения, что определяется ниже для каждого типа сообщения.

6.2.2.2 Поле «Основная версия формата данных» (Major)

Значения поля «Основная версия формата данных» (Major) и их описание приведены в таблице 2.

Таблица 2 — Значения поля «Основная версия формата данных» (Major)

Значение поля	Описание
00	LoRaWAN RU
01..11	RFU

Примечание — Значения поля «Основная версия формата данных» (Major) определяют формат сообщений, которыми обмениваются в ходе процедуры присоединения к сети (активации) (см. 6.4.2), и первые четыре байта поля «MAC-сообщение» (MACPayload). Для каждой основной версии формата данных оконечные устройства могут реализовывать разные неосновные версии формата данных. Неосновная версия, используемая оконечным устройством, должна быть известна сетевому серверу до ее использования (например, как часть информации, персонализирующей устройство). Если устройство или сервер сети получают данные неизвестной или неподдерживаемой версии формата данных, то они должны быть проигнорированы.

6.2.3 Поле «MAC-сообщение» (MACPayload)

Структура поля «MAC-сообщение» (MACPayload) приведена на рисунке 9 и содержит поле «Заголовок MAC-сообщения» (FHDR), необязательное поле «Порт» (FPort) и необязательное поле «Прикладные данные» (FRMPayload).

Размер (в байтах)	7…22	0…1	0…N
MAC-сообщение (MACPayload)	Заголовок MAC-сообщения (FHDR)	Порт (FPort)	Прикладные данные (FRMPayload)

Рисунок 9 — Структура поля «MAC-сообщение» (MACPayload)

Поле «Прикладные данные» (FRMPayload) имеет размер 0…N байт, определяемый согласно региональным параметрам 9.1.6.

Поле «Заголовок MAC-сообщения» (FHDR) предназначено для адресации оконечных устройств и описано в 6.2.3.1.

Поле «Порт» (FPort) предназначено для адресации поля «Прикладные данные» (FRMPayload) на уровне устройства и описано в 6.2.3.2.

Поле «Прикладные данные» (FRMPayload) предназначено для передачи данных по целевому назначению устройства.

Поле «MAC-сообщение» (MACPayload), состоящее только из поля «Заголовок MAC-сообщения» (FHDR), является корректным.

6.2.3.1 Поле «Заголовок MAC-сообщения» (FHDR)

Структура поля «Заголовок MAC-сообщения» (FHDR) приведена на рисунке 10.

Размер (в байтах)	4	1	2	0..15
Заголовок MAC-сообщения (FHDR)	Короткий адрес оконечного устройства (DevAddr)	Управление кадром (FCtrl)	Счетчик кадров (FCnt)	Параметры кадра (FOpts)

Рисунок 10 — Структура поля «Заголовок MAC-сообщения» (FHDR)

Структура поля «Управление кадром» (FCtrl) для нисходящих сообщений приведена на рисунке 11.

Номер бита	7	6	5	4	[3..0]
Управление кадром (FCtrl)	Адаптивная скорость передачи данных (ADR)	RFU	Подтверждение получения сообщения (ACK)	Отложенные кадры (FPending)	Длина параметров кадра (FOptsLen)

Рисунок 11 — Структура поля «Управление кадром» (FCtrl) для нисходящих сообщений

Структура поля «Управление кадром» (FCtrl) для восходящих сообщений приведена на рисунке 12.

Номер бита	7	6	5	4	[3..0]
Управление кадром (FCtrl)	Адаптивная скорость передачи данных (ADR)	Запрос подтверждения адаптивной скорости передачи данных (ADRACKReq)	Подтверждение получения сообщения (ACK)	RFU	Длина параметров кадра (FOptsLen)

Рисунок 12 — Структура поля «Управление кадром» (FCtrl) для восходящих сообщений

а) Адаптивное управление скоростью передачи данных [поля «Адаптивная скорость передачи данных» (ADR), «Запрос подтверждения адаптивной скорости передачи данных (ADRACKReq)]

Сеть LoRaWAN RU позволяет оконечным устройствам индивидуально настраивать любые из допустимых скоростей передач данных и выходную мощность передатчика. Эта особенность использу-

ется в протоколе LoRaWAN RU для адаптации и оптимизации скорости передачи данных и выходной мощности передатчика стационарных и малоподвижных оконечных устройств. Для указания данного свойства используется поле «Адаптивная скорость передачи данных» (ADR), и в этом случае сеть будет оптимизирована для использования самой быстрой возможной скорости передачи данных.

Адаптивное управление скоростью передачи данных становится невозможным, когда затухание сигнала в радиоканале быстро и постоянно меняется. Когда сетевой сервер не в состоянии управлять скоростью передачи данных устройства, управление должно осуществляться на уровне приложения оконечного устройства. В этом случае рекомендуется использовать различные скорости передачи данных. Уровень приложения должен всегда минимизировать общее эфирное время с учетом состояния сети.

Если значение поля «Адаптивная скорость передачи данных» (ADR) восходящего сообщения установлено, то сеть будет управлять скоростью передачи данных и выходной мощностью передатчика оконечного устройства через соответствующие MAC-команды. Если значение поля «Адаптивная скорость передачи данных» (ADR) не установлено, то сеть не будет управлять скоростью передачи данных и выходной мощностью передатчика оконечных устройств независимо от качества принимаемого сигнала. Дополнительно с целью снижения количества потерянных пакетов сервер сети может посылать команды, изменяющие маску канала или количество повторений для каждого восходящего сообщения.

Если значение поля «Адаптивная скорость передачи данных» (ADR) нисходящего сообщения установлено, то оно информирует оконечное устройство, что сетевой сервер может посылать ADR-команды. Оконечное устройство может устанавливать/снимать значение поля «Адаптивная скорость передачи данных» (ADR) восходящего сообщения.

Если значение поля «Адаптивная скорость передачи данных» (ADR) нисходящего сообщения не установлено, то для оконечного устройства это означает, что из-за быстрых изменений в радиоканале сеть временно не может оценить лучшую скорость передачи данных. В этом случае у устройства есть следующий выбор:

- сбросить значение поля «Адаптивная скорость передачи данных» (ADR) восходящего сообщения и управлять своей скоростью передачи данных в восходящей линии связи с использованием собственной стратегии. Эта стратегия должна быть типовой для мобильных оконечных устройств;
- игнорировать (сохранить поле «Адаптивная скорость передачи данных» (ADR) восходящего сообщения установленным) и применить нормальную сниженную скорость передачи данных при отсутствии нисходящих ADR-команд. Эта стратегия должна быть типовой для неподвижных оконечных устройств.

Значение поля «Адаптивная скорость передачи данных» (ADR) может быть установлено и сброшено оконечным устройством или сетью по запросу. Однако по мере возможности ADR-схема должна быть включена, чтобы увеличить время автономной работы оконечного устройства и увеличить производительность сети.

Примечание — В некоторых случаях мобильные оконечные устройства большую часть времени являются неподвижными. Поэтому в зависимости от состояния мобильности оконечное устройство может запросить сеть оптимизировать скорость передачи данных, используя значение поля «Адаптивная скорость передачи данных» (ADR) восходящего сообщения.

По умолчанию выходная мощность передатчика равна максимальной допустимой мощности передачи для устройства, с учетом ограничений, описанных в разделе 9. Устройство должно использовать этот уровень мощности, пока не будет запроса об уменьшении от сети через MAC-команду LinkADRReq.

Если скорость передачи данных устройства оптимизирована сетью и превышает скорость передачи данных устройства по умолчанию или выходная мощность передатчика ниже, чем по умолчанию, то устройство должно периодически проверять, что сеть по-прежнему получает восходящие пакеты. Каждый раз, когда увеличивается счетчик восходящих кадров (для каждого нового восходящего сообщения при повторных передачах счетчик не увеличивается), устройство увеличивает значение счетчика ADR_ACK_CNT. После превышения счетчиком ADR_ACK_CNT (ADR_ACK_CNT >= ADR_ACK_LIMIT) порогового значения ADR_ACK_LIMIT восходящих сообщений без какого-либо ответа сервера сети, устройство устанавливает значение поля «Запрос подтверждения адаптивной скорости передачи данных» (ADRACKReq). Сервер сети должен отреагировать нисходящим кадром в течение следующих ADR_ACK_DELAY кадров (восходящих), любой полученный нисходящий пакет сбрасывает счетчик ADR_ACK_CNT восходящей линии связи. Значение поля «Подтверждение получения сообщения» (ACK) в нисходящем сообщении устанавливать не нужно, так как любой ответ в окно приема оконечно-

го устройства указывает на то, что шлюз (базовая станция) все еще получает восходящие сообщения от этого устройства. Если ответ не будет получен в течение ближайших ADR_ACK_DELAY восходящих сообщений (т. е. после превышения количества восходящих сообщений, оставшихся без ответа со стороны сервера сети, более чем ADR_ACK_LIMIT + ADR_ACK_DELAY), то устройство должно попытаться восстановить связь путем наращивания выходной мощности передатчика до значения по умолчанию, если это возможно, и затем переключиться на более низкую скорость передачи данных, что увеличивает дальность связи. Оконечное устройство должно продолжать снижать свою скорость передачи данных шаг за шагом, всякий раз при достижении ADR_ACK_DELAY. После того как устройство достигнет самой низкой скорости передачи данных, оно должно повторно включить все частотные каналы восходящей линии связи по умолчанию.

Значение поля «Запрос подтверждения адаптивной скорости передачи данных» (ADRACKReq) не должно быть установлено, если устройство использует скорость передачи данных и выходную мощность передатчика по умолчанию, потому что в этом случае никакие действия не могут быть предприняты для улучшения качества связи.

Примечания

1 Отсутствие необходимости немедленного ответа на запрос подтверждения ADR обеспечивает гибкость сети при оптимальном планировании своих передач по нисходящей линии связи.

2 При передаче по восходящей линии связи значение поля «Запрос подтверждения адаптивной скорости передачи данных» (ADRACKReq) устанавливается, если ADR_ACK_CNT >= ADR_ACK_LIMIT и текущая скорость передачи данных больше, чем определенная для устройства минимальная скорость передачи данных, или мощность передачи меньше, чем по умолчанию, или текущая маска канала использует только подмножество всех каналов по умолчанию. При других условиях он обнуляется.

В таблице 3 приведен пример обратной последовательности снижения скорости передачи данных при условии, что константы ADR_ACK_LIMIT=32 и ADR_ACK_DELAY=32.

Таблица 3 — Пример обратной последовательности снижения скорости передачи данных

Значение счетчика ADR_ACK_CNT	Поле «Запрос подтверждения адаптивной скорости передачи данных» (ADRACKReq)	Скорость передачи данных	Выходная мощность передатчика (дБм)	Маска канала
От 0 до 63	0	SF11	9	Текущий список каналов
От 64 до 95	1	SF11	9	Текущий список каналов
От 96 до 127	1	SF11	14	Текущий список каналов
От 128 до 159	1	SF12	14	Текущий список каналов
Более 160	0	SF12	14	Все доступные каналы

б) Поле «Подтверждение получения сообщения» (ACK) и процедура подтверждения

При получении сообщения, требующего уведомления о получении данных, получатель должен ответить сообщением, в котором установлено значение поля «Подтверждение получения сообщения» (ACK). Если отправителем является оконечное устройство, сеть попытается отправить подтверждение, используя одно из окон приема, открытое оконечным устройством после операции отправки. Если отправителем является шлюз, оконечное устройство передает уведомление по своему усмотрению (см. примечание ниже).

Подтверждение отправляется только в ответ на последнее полученное сообщение и никогда не ретранслируется.

Примечание — Оконечному устройству разрешено, насколько возможно, упростить процедуру подтверждения и иметь несколько вариантов ее выполнения. Оконечное устройство может передавать явное (возможно, пустое) сообщение с подтверждением получения данных сразу после получения сообщения с данными, требующего подтверждения получения. Кроме того, оконечное устройство может отложить передачу подтверждения, прикрепив его к следующему сообщению данных.

в) Процедура повторной передачи

Нисходящий кадр (требующий или не требующий подтверждения от оконечного устройства) не должен повторно отправляться сервером сети с использованием одного и того же счетчика значения нисходящих кадров. Если после отправки нисходящего сообщения, требующего подтверждения, сервер сети не получил от устройства уведомления о доставке, то сервер сети должен уведомить об этом сервер приложения. Сервер приложения принимает решение о целесообразности повторной передачи нисходящего сообщения, требующего подтверждения.

Восходящие кадры (требующие и не требующие подтверждения) передаются число раз, равное NbTrans (см. 6.3.3), за исключением, если получено корректное нисходящее сообщение в ходе одной следующей передачи. Параметр NbTrans может использоваться администратором сети, чтобы контролировать избыточность восходящих пакетов узла связи для получения заданного качества обслуживания. Оконечное устройство должно выполнять скачкообразное переключение частоты между повторными передачами. После каждого повторения оно должно ждать, пока не закроется окно приема. Задержка между повторными передачами остается на усмотрение оконечного устройства и может быть различной для каждого оконечного устройства.

Устройство должно остановить любую дальнейшую ретрансляцию восходящих кадров, требующих подтверждения, если получен соответствующий нисходящий кадр с подтверждением.

Устройства класса С должны прекратить любую дальнейшую ретрансляцию восходящих кадров, когда корректное нисходящее сообщение получено в окно приема RX1.

Устройства класса А должны прекратить любую дальнейшую ретрансляцию восходящих кадров, когда корректное нисходящее сообщение получено в окно приема RX1 или RX2.

Если сервер сети получает один и тот же восходящий кадр более NbTrans раз, то это может быть признаком атаки на сервер или неисправности устройства. В этом случае сервер сети не должен обрабатывать избыточные кадры.

Примечание — Сервер сети, обнаружив атаку в виде повторных передач, может принять дополнительные меры, такие как уменьшение параметра NbTrans на 1 или удаление дублирующих восходящих кадров, принятых через один и тот же канал, или другие механизмы.

г) Поле «Отложенные кадры» (FPending) только для нисходящих сообщений

Поле «Отложенные кадры» (FPending) используется только в нисходящей линии связи и указывает на то, что сеть имеет данные, ожидающие своей отправки, и поэтому запрашивает оконечное устройство максимально быстро открыть еще одно окно приема посредством отправки другого восходящего сообщения.

Подробное использование поля «Отложенные кадры» (FPending) описано в 8.3.

д) Поле «Счетчик кадров» (FCnt)

Каждое оконечное устройство имеет три счетчика кадров, чтобы отслеживать и хранить число кадров данных, переданных в восходящую линию связи сетевому серверу (FCntUp) и отправленных в нисходящую линию связи сервером сети на оконечное устройство (AFCntDown и NFCntDown).

В нисходящем направлении существуют две разные схемы счетчиков кадров:

- единая схема счетчика, в котором все порты используют один и тот же счетчик кадров AFCntDown = NFCntDown = FCntDown, когда оконечное устройство работает по спецификации LoRaWAN v.1.0;

- схема с двумя счетчиками, в которой первый счетчик NFCntDown используется для MAC-команд на порт 0 или когда поле FPort отсутствует. Второй счетчик AFCntDown используется для всех других портов, когда устройство работает по спецификации LoRaWAN v.1.1.

В схеме с двумя счетчиками счетчик NFCntDown управляется сервером сети, а счетчик AFCntDown управляется сервером приложений.

Примечание — Спецификации LoRaWAN v.1.0 и более ранние версии поддерживают только один счетчик FCntDown (общий по всем портам), и сетевой сервер должен обеспечить поддержку схемы для устройств, работающих по спецификации LoRaWAN с версией до 1.1.

Всякий раз, когда устройство с активацией по воздуху (OTAA) успешно обрабатывает сообщение подтверждения присоединения (переприсоединения) к сети (Join-Accept), счетчик кадров на устройстве (FCntUp) и счетчики кадров на стороне сети (NFCntDown и AFCntDown) для этого оконечного устройства сбрасываются до 0.

Устройства с активацией через персонализацию (ABP, Activation By Personalization) имеют свои счетчики кадров, которые инициализируются в 0 при изготовлении. На устройствах с активацией через персонализацию счетчики кадров не должны сбрасываться на протяжении времени жизни устройства. Если оконечное устройства подвергается отключению питания во время срока службы (например, замена аккумуляторной батареи), счетчики кадров должны сохраняться.

Впоследствии счетчик кадров FCntUp увеличивается с каждым восходящим сообщением. Счетчик кадров NFCntDown увеличивается с каждым нисходящим сообщением в случае, когда значение в поле «Порт» (FPort) равно нулю или данное поле отсутствует. Счетчик кадров AFCntDown увеличивается с каждым нисходящим сообщением при значении поля «Порт» (FPort), отличном от нуля. На стороне получателя соответствующий счетчик будет синхронизироваться с полученным значением при условии, что полученное значение было увеличено по сравнению с текущим значением счетчика и поле «Код целостности сообщения» (MIC) соответствует значению кода целостности сообщения, вычисленному локально с использованием соответствующего сетевого сеансового ключа. Счетчик FCnt не увеличивается при многократных передачах кадров, требующих и не требующих подтверждения получения (см. параметр NbTrans). Сетевой сервер должен отбрасывать прикладные данные из повторно принятых кадров и передавать серверу приложений только один экземпляр кадра.

Счетчики кадров имеют размерность 32 бита. В поле «Счетчик кадров» (FCnt) передаются только младшие 16 бит из 32-битного счетчика кадров (т. е. от счетчика FCntUp для кадров данных, передаваемых в восходящую линию связи, и счетчика AFCntDown/NFCntDown для кадров данных, передаваемых по нисходящей линии).

Примечание — Порядок следования байтов — «от младшего к старшему» (little-endian).

Оконечное устройство не должно использовать те же значения счетчика данных FCntUp с одним и тем же сеансовым ключом приложения или сеансовым сетевым ключом, за исключением случаев ретрансляции одного и того же кадра с подтверждением или без подтверждения.

Оконечное устройство не должно обрабатывать любые повторные передачи нисходящих кадров. Последующие повторные передачи должны игнорироваться без обработки.

Примечания

1 Это означает, что устройство будет отправлять подтверждение единожды, только после приема нисходящего кадра, требующего подтверждения. Аналогично устройство будет генерировать только одно восходящее сообщение после приема кадра с установленным битом поля «Отложенные кадры» (FPending).

2 Поскольку поле «Счетчик кадров» (FCnt) несет только младшие 16 бит из 32-битного счетчика кадров, сервер должен вычислить 16 старших бит счетчика кадров из наблюдений за трафиком.

е) Параметры кадра (поле «Длина параметров кадра» (FOptsLen) и поле «Параметры кадра» (FOpts))

Поле «Длина параметров кадра» (FOptsLen) в поле «Управление кадром» (FCtrl) указывает фактическую длину поля «Параметры кадра» (FOpts), включенного в кадр.

Поле «Параметры кадра» (FOpts) передает MAC-команды длиной до 15 байт, которые присоединяются к кадру данных. Список MAC-команд приведен в 6.3.

Если значение поля «Длина параметров кадра» (FOptsLen) равно нулю, то поле «Параметры кадра» (FOpts) должно отсутствовать. Если значение поля «Длина параметров кадра» (FOptsLen) отличается от нуля, т. е. если MAC-команды присутствуют в поле «Параметры кадра» (FOpts), то не может быть использовано нулевое значение в поле «Порт» (FPort) (поле должно отсутствовать или значение отличаться от нуля).

MAC-команды не могут одновременно присутствовать в поле «Прикладные данные» (FRMPayload) и в поле «Параметры кадра» (FOpts). Если это произойдет, то устройство должно игнорировать кадр.

Если в заголовке кадра имеется поле «Параметры кадра» (FOpts), то поле «Параметры кадра» (FOpts) должно быть закодировано до того, как будет рассчитано значение поля «Код целостности сообщения» (MIC).

6.2.3.2 Поле «Порт» (FPort)

Если поле «Прикладные данные» (FRMPayload) не является пустым, то поле «Порт» (FPort) должно присутствовать. При наличии поля «Порт» (FPort) значение, равное 0, указывает, что поле «Прикладные данные» (FRMPayload) содержит только MAC-команды, и любые полученные кадры в этом случае будут обработаны на уровне реализации LoRaWAN RU (список допустимых MAC-команд приведен в 6.3). Значения поля «Порт» (FPort) в диапазоне от 1 до 223 (от 0x01 до 0xDF) выделены под приложения, и любые полученные кадры в этом случае должны быть доступны на уровне приложения. Значение поля «Порт» (FPort), равное 224, выделено под протокол испытаний LoRaWAN RU уровня MAC. Реализация LoRaWAN RU должна стирать любые запросы передачи от уровня приложения, где значение поля «Порт» (FPort) не входит в диапазон от 1 до 224.

Примечание — Значение поля «Порт» (FPort), равное 224, предназначено для сценариев беспроводных испытаний соответствия окончательной версии устройства MAC настоящей спецификации, без необходимости полагаться на определенные испытательные версии устройств для прикладных задач. Испытание не должно совпадать со штатными операциями, но реализация MAC-уровня устройства должна быть точно той, какая используется для обычного приложения

Протокол испытания работает на прикладном уровне и определяется за рамками настоящего стандарта, так как он является протоколом прикладного уровня.

Значения поля «Порт» (FPort) в диапазоне от 225 до 255 (от 0xE1 до 0xFF) зарезервированы для будущих стандартизированных расширений приложений.

Структура поля «MAC-сообщение» (MACPayload) приведена на рисунке 13.

Размер (в байтах)	От 7 до 22	0 или 1	От 0 до *N*
MAC-сообщение (MACPayload)	Заголовок MAC-сообщения (FHDR)	Порт (FPort)	Прикладные данные (FRMPayload)

Рисунок 13 — Структура поля «MAC-сообщение» (MACPayload)

N — число байтов прикладных данных. Допустимый диапазон *N* является региональным параметром и определяется в разделе 9. *N* должно быть равным или меньшим, чем:

$$N \leq M - 1 \text{ — (длина поля «Заголовок MAC-сообщения» (FHDR) в байтах),}$$

где *M* — максимальная длина данных в поле «MAC-сообщение» (MACPayload).

6.3 MAC-команды

Набор MAC-команд предназначен для сетевого администрирования и может быть использован для обмена между сервером сети и оконечным устройством на MAC-уровне (см. таблицу 4). Команды MAC-уровня не обрабатываются сервером приложений и приложением, запущенным на устройстве.

Один кадр данных может содержать любую последовательность MAC-команд, вставленную в поле «Параметры кадра» (FOpts) или отправленную в отдельном кадре данных, в поле «Прикладные данные» (FRMPayload) со значением поля «Порт» (FPort), равным 0.

MAC-команды, передаваемые в поле «Параметры кадра» (FOpts), отправляются в кодированном виде и не должны превышать 15 байт. MAC-команды, отправляемые в поле «Прикладные данные» (FRMPayload), всегда кодируются, и их длина не должна превышать максимальную длину поля «Прикладные данные» (FRMPayload).

MAC-команда состоит из поля «Идентификатор команды» (CID) размером 1 байт и поля «Атрибуты команды» размером от 0 до 14 байт. Для некоторых команд поле «Атрибуты команды» может быть пустым.

MAC-команды со значениями идентификаторов команды (CID) от 0x01 до 0x7F предназначены для использования во всех сетях LoRaWAN RU.

Приемная сторона отвечает/подтверждает получение MAC-команд в том же порядке, как они передаются. Ответ для каждой MAC-команды последовательно добавляется в буфер. На все MAC-команды, полученные в одном кадре, ответы должны быть переданы в одном кадре (т. е. буфер, содержащий ответы на MAC-команды, должен быть отправлен в одном кадре). Если длина буфера с MAC-ответами больше, чем максимальная длина поля «Параметры кадра» (FOpts), устройство должно отправить буфер в поле «Прикладные данные» (FRMPayload) на порт 0. Если устройству надо отпра-

вить прикладные данные и MAC-ответы и они не помещаются в один кадр, то MAC-ответы должны быть отправлены в первую очередь. Если длина буфера превышает максимальный используемый размер поля «Прикладные данные» (FRMPayload), устройство перед сборкой кадра должно уменьшить буфер до максимального размера поля «Прикладные данные» (FRMPayload). Поэтому ответы на последние MAC-команды могут быть неполными. В любом случае полный список MAC-команд выполняется, даже если буфер, содержащий MAC-ответы, должен быть обрезан. Сетевой сервер не должен генерировать последовательность MAC-команд, на которые оконечное устройство не может ответить одним восходящим кадром. Сетевой сервер должен вычислять максимальный размер поля «Прикладные данные» (FRMPayload) для ответов на MAC-команды следующим образом:

- если поле «Адаптивная скорость передачи данных» (ADR) последнего восходящего сообщения имеет значение 0, то необходимо устанавливать максимальный размер поля «Прикладные данные» (FRMPayload), соответствующий самой низкой скорости передачи данных;
- если поле «Адаптивная скорость передачи данных» (ADR) последнего восходящего сообщения имеет значение 1, то необходимо устанавливать максимальный размер поля «Прикладные данные» (FRMPayload) соответствующим скорости передачи данных, используемой устройством для передачи последнего восходящего сообщения.

Примечание — При получении обрезанного MAC-ответа сервер сети может ретранслировать MAC-команды, на которые не получил ответ.

Таблица 4 — MAC-команды

CID	Команда	Передается		Краткое описание
		оконечному устройству	базовой станции	
0x01	ResetInd	x	—	Используются устройствами с активацией через персонализацию для индикации сброса и согласования версий протокола
0x01	ResetConf	—	x	Подтверждает команду *ResetInd*
0x02	LinkCheckReq	x	—	Используется оконечным устройством для проверки его подключения к сети
0x02	LinkCheckAns	—	x	Ответ на команду *LinkCheckReq*. Содержит оценку мощности полученного сигнала, указывающую на качество приема оконечного устройства (устойчивость связи)
0x03	LinkADRReq	—	x	Запрашивает оконечное устройство изменить скорость передачи данных, мощность передачи, количество повторений или канал
0x03	LinkADRAns	x	—	Подтверждает команду *LinkADRReq*
0x04	DutyCycleReq	—	x	Устанавливает максимальное агрегированное значение рабочего цикла устройства на передачу
0x04	DutyCycleAns	x	—	Подтверждает команду *DutyCycleReq*
0x05	RXParamSetupReq	—	x	Устанавливает параметры окон приема
0x05	RXParamSetupAns	x	—	Подтверждает команду *RXParamSetupReq*
0x06	DevStatusReq	—	x	Запрашивает статус оконечного устройства
0x06	DevStatusAns	x	—	Возвращает статус (состояние) оконечного устройства, а именно уровень заряда его батареи и отношение сигнал/шум (оценка демодуляции)
0x07	NewChannelReq	—	x	Создает или изменяет определение радиоканала
0x07	NewChannelAns	x	—	Подтверждает команду *NewChannelReq*

Окончание таблицы 4

CID	Команда	Передается		Краткое описание
		оконечному устройству	базовой станции	
0x08	RXTimingSetupReq	—	x	Устанавливает временные интервалы для окон приема
0x08	RXTimingSetupAns	x	—	Подтверждает команду *RXTimingSetupReq*
0x09	TxParamSetupReq	—	x	Используется сервером сети, чтобы установить максимально допустимое время задержки и максимальную эффективную изотропную мощность излучения (EIRP) оконечного устройства на основе локальных соглашений и нормативных актов
0x09	TxParamSetupAns	x	—	Подтверждает команду *TxParamSetupReq*
0x0A	DlChannelReq	—	x	Изменяет определение нисходящего радиоканала окна приема RX1 путем смещения частоты передачи нисходящей линии от частоты восходящей линии связи (т. е. создание асимметричного канала)
0x0A	DlChannelAns	x	—	Подтверждает команду *DlChannelReq*
0x0B	RekeyInd	x	—	Используется устройством с активацией по воздуху для оповещения об обновлении сеанса связи устройства с сервером сети (обновление ключей)
0x0B	RekeyConf	—	x	Подтверждает команду *RekeyInd*
0x0C	ADRParamSetupReq	—	x	Используется сервером сети для установки ADR_ACK_LIMT и ADR_ACK_DELAY параметров оконечного устройства
0x0C	ADRParamSetupAns	x	—	Подтверждает команду *ADRParamSetupReq*
0x0D	DeviceTimeReq	x	—	Используется оконечным устройством для запроса текущей даты и времени
0x0D	DeviceTimeAns	—	x	Сеть отправляет ответ на запрос *DeviceTimeReq*
0x0E	ForceRejoinReq	—	x	Посылается сетью для запроса немедленного переприсоединения устройства к сети, дополнительно указывается количество и периодичность повторов
0x0F	RejoinParamSetup Req	—	x	Используется сетью для установки периодичности отправки устройством запросов на переприсоединение к сети
0x0F	RejoinParamSetup Ans	x	—	Подтверждает команду *RejoinParamSetupReq*
От 0x80 до 0xFF	Проприетарные команды	x	x	Зарезервировано для команд, действующих только в региональных сетях

Примечания

1 В основном оконечное устройство будет отвечать только один раз на любую полученную MAC-команду. Если ответ потерян, то сеть вынуждена будет снова послать команду. Сервер сети решает, что команда должна быть отправлена повторно, когда она получает новое восходящее сообщение, которое не содержит ответа.

Только *RxParamSetupReq*, *RxTimingSetupReq* и *DlChannelReq* имеют другой механизм подтверждения, описанный в соответствующих разделах, так как они влияют на параметры нисходящей линии связи.

2 Когда MAC-команда инициируется оконечным устройством, сеть делает все возможное для отправки подтверждения/ответа в окна приема RX1/RX2 сразу после запроса. Если ответ не получен в окна RX1 и RX2, то оконечное устройство может реализовать любой механизм повтора, который сочтет нужным.

3 Длина MAC-команды не задается явно и должна быть неявно известной по MAC-реализации. Поэтому неизвестные MAC-команды не могут быть пропущены, и первая неизвестная MAC-команда завершает обработку последовательности MAC-команд. Целесообразно использовать MAC-команды, соответствующие версиям стандарта, в котором MAC-команда была впервые опубликована. Таким образом, все MAC-команды, реализованные до появления настоящего стандарта, могут быть обработаны оконечным устройством даже среди MAC-команд, описанных только в текущей версии спецификации LoRaWAN RU, и если она новее, чем реализация оконечного устройства.

6.3.1 Команды индикации сброса (*ResetInd, ResetConf*)

Данная MAC-команда доступна только для устройств с активацией через персонализацию, активированных в сети с сервером сети, поддерживающим LoRaWAN v.1.1. На сервере сети, поддерживающем только LoRaWAN v.1.0, данная MAC-команда не реализована.

Устройства с активацией по воздуху не должны отправлять эту команду. Сетевой сервер должен игнорировать команду *ResetInd*, поступившую от устройства с активацией по воздуху.

С помощью команды *ResetInd* устройство с активацией через персонализацию извещает сети, что оно было повторно инициализировано и что оно переключено на свои MAC- и радионастройки по умолчанию (т. е. параметры, изначально запрограммированные в устройстве при изготовлении, за исключением трех счетчиков кадров). Команда *ResetInd* должна добавляться в поле «Параметры кадра» (FOpts) всех восходящих кадров, пока не будет получена команда *ResetConf*.

Данная команда не является сигналом сетевому серверу, что были сброшены счетчики кадров. Счетчики кадров нисходящих и восходящих сообщений не должны сбрасываться в устройствах с активацией через персонализацию.

Примечание — Данная команда предназначена для устройств с активацией через персонализацию, питание которых может быть отключено в какой-то момент времени (например, замена батареи). Устройство может потерять настройки соединения на сеансном уровне, хранящиеся в ОЗУ (кроме счетчиков кадров, которые должны быть сохранены в энергонезависимой памяти). В этом случае устройство нуждается в том, чтобы как-то сообщить серверу сети о потере настроек соединения на сеансном уровне. В будущих версиях протокола LoRaWAN RU эта команда может также использоваться для согласования некоторых параметров протокола между устройством и сервером сети.

Команда *ResetInd* включает в себя дополнительный номер версии LoRaWAN RU, поддерживаемой устройством. Структура команды *ResetInd* приведена на рисунке 14.

Размер (в байтах)	1
Данные *ResetInd* (*ResetInd* Payload)	Версия LoRaWAN RU устройства (Dev LoRaWAN RU version)

Рисунок 14 — Структура команды *ResetInd*

Структура поля «Версия LoRaWAN RU устройства» (Dev LoRaWAN RU version) команды *ResetInd* приведена на рисунке 15.

Биты	7:4	3:0
Версия LoRaWAN RU устройства (Dev LoRaWAN RU version)	RFU	Дополнительный номер версии (Minor)=1

Рисунок 15 — Структура поля «Версия LoRaWAN RU устройства» (Dev LoRaWAN RU version) команды *ResetInd*

Поле «Дополнительный номер версии» (Minor) указывает на дополнительный номер версии LoRaWAN RU, поддерживаемый оконечным устройством (см. таблицу 5).

Таблица 5 — Значения поля «Дополнительный номер версии» (Minor)

Дополнительный номер версии (Minor)	Значение поля
RFU	0
1 (LoRaWAN RU x.1)	1
RFU	От 2 до 15

Когда сетевой сервер получает *ResetInd*, он отвечает командой *ResetConf*.

Команда ResetConf содержит один байт данных, закодированных в соответствии с версией LoRaWAN RU, поддерживаемой сервером сети с использованием формата, соответствующего версии LoRaWAN RU устройства (Dev LoRaWAN RU version). Структура команды *ResetInd* приведена на рисунке 16.

Размер (в байтах)	1
Данные ResetConf (ResetConf Payload)	Версия LoRaWAN RU сервера сети (Serv LoRaWAN RU version)

Рисунок 16 — Структура команды *ResetConf*

Версия сервера, которую несет *ResetConf*, должна совпадать с версией устройства. Любое другое значение является недопустимым.

Если версия сервера не совпадает с версией устройства, устройство должно отбросить команду *ResetConf* и повторно отправить команду *ResetInd* в следующем восходящем кадре.

6.3.2 Команды проверки подключения к сети (*LinkCheckReq, LinkCheckAns*)

С помощью команды *LinkCheckReq* оконечное устройство может проверить свое подключение к сети. Команда не имеет полезных данных.

Когда сетевой сервер получает *LinkCheckReq* через один или несколько шлюзов, он отвечает командой *LinkCheckAns*. Структура команды *LinkCheckAns* приведена на рисунке 17.

Размер (в байтах)	1	1
Данные LinkCheckAns (LinkCheckAns Payload)	Устойчивость демодуляции (Margin)	Число шлюзов (GwCnt)

Рисунок 17 — Структура команды *LinkCheckAns*

Поле «Устойчивость демодуляции» (Margin) представляет собой 8-битовое целое число без знака в диапазоне от 0 до 254 и указывает значение устойчивости связи в дБ, полученное по факту успешного приема последней MAC-команды *LinkCheckReq*. Значение, равное 0, означает, что кадр был получен на минимальном уровне демодуляции (0 дБ или при отсутствии значения), а значение, равное 20, например, означает, что кадр достиг шлюза с 20 дБ запаса относительно порога демодуляции. Значение, равное 255, зарезервировано для будущего использования.

Поле «Число шлюзов» (GwCnt) определяет число шлюзов (базовых станций), которые успешно получили последнюю команду *LinkCheckReq*.

Значения минимального уровня соотношения сигнал/шум для демодуляции пакета приведены в таблице 6.

Таблица 6 — Значения минимального уровня соотношения сигнал/шум для демодуляции пакета

Скорость передачи	Сигнал/шум (SNR) минимального уровня демодуляции пакета (дБ)
DR0	−20
DR1	−17,5
DR2	−15
DR3	−12,5
DR4	−10
DR5	−7,5

6.3.3 Адаптация скорости передачи данных (*LinkADRReq, LinkADRAns*)

С помощью команды *LinkADRReq* сетевой сервер запрашивает оконечное устройство выполнить адаптацию скорости передачи данных. Структура команды *LinkADRReq* приведена на рисунке 18.

Размер (в байтах)	1	2	1
Данные LinkADRReq (LinkADRReq Payload)	Запрошенная скорость передачи данных и выходная мощность передатчика (DataRate_TXPower)	Маска канала (ChMask)	Избыточность (Redundancy)

Рисунок 18 — Структура команды *LinkADRReq*

Структура поля «Запрошенная скорость передачи данных и выходная мощность передатчика» (DataRate_TXPower) команды *LinkADRReq* приведена на рисунке 19.

Биты	[7:4]	[3:0]
Запрошенная скорость передачи данных и выходная мощность передатчика (DataRate_TXPower)	Запрошенная скорость передачи данных (DataRate)	Выходная мощность передатчика (TXPower)

Рисунок 19 — Структура поля «Запрошенная скорость передачи данных и выходная мощность передатчика» (DataRate_TXPower) команды *LinkADRReq*

Запрошенная скорость передачи данных (DataRate) и выходная мощность передатчика (TXPower) являются региональными параметрами и определены в разделе 9. Выходная мощность передатчика, указанная в команде, должна рассматриваться как максимальная мощность передачи, на которой может работать устройство. Если в команде задана мощность, превышающая максимальную мощность устройства, то оконечное устройство должно подтвердить успешное выполнение команды и работать на своей максимально возможной мощности.

Значение 0xF (15 в десятичном формате) запрошенной скорости передачи данных (DataRate) или выходной мощности передатчика (TXPower) означает, что устройство должно игнорировать это поле и сохранить текущее значение параметра. Маска канала (ChMask) кодирует использование каналов для передачи в восходящей линии связи (бит 0 соответствует младшему разряду) в соответствии с таблицей 7.

Таблица 7 — Кодирование поля «Маска канала» (ChMask)

Биты	Используемые каналы
0	Канал 1
1	Канал 2
…	…
15	Канал 16

Бит в поле «Маска канала» (ChMask), установленный в 1, означает, что соответствующий канал может использоваться для передачи в восходящей линии связи, если этот канал обеспечивает скорость передачи данных, в настоящее время используемую устройством. Бит, установленный в 0, означает, что соответствующие каналы следует исключить.

Поле «Избыточность» (Redundancy) (см. рисунок 20) включает поле «Число передач» (NbTrans). Это поле используется для восходящих кадров, требующих и не требующих подтверждения получения. Значением по умолчанию является 1, которое соответствует одной передаче каждого кадра. Допустимый диапазон от 1 до 15. Если получено значение поля «Число передач» (NbTrans), равное 0, то устройство должно сохранить текущее значение числа передач неизменным.

Биты	7	[6:4]	[3:0]
Избыточность (Redundancy)	RFU	Управление маской канала (ChMaskCntl)	Число передач (NbTrans)

Рисунок 20 — Структура поля «Избыточность» (Redundancy)

Поле «Управления маской канала» (ChMaskCntl) управляет интерпретацией ранее определенного битового поля «Маска канала» (ChMask). Поле «Управления маской канала» (ChMaskCntl) контролирует блок из 16 каналов, к которым применяется поле «Маска канала» (ChMask). Оно также может быть использовано, чтобы глобально включить или выключить все каналы. Использование этого атрибута является региональным параметром и определяется в разделе 9.

Сетевой сервер может включить несколько последовательных команд *LinkAdrReq* в одно нисходящее сообщение. С целью конфигурирования маски канала оконечного устройства оконечное устройство должно обработать всю последовательность *LinkAdrReq* в сообщениях в том порядке, в котором они переданы в нисходящем сообщении, как один единый блок команд. Сетевой сервер не должен включать в нисходящее сообщение более одного такого блока команд. Оконечное устройство должно послать одну команду *LinkAdrAns*, чтобы подтвердить или отклонить весь единый ADR командный блок. Если нисходящее сообщение несет более одного единого ADR командного блока, то устройство должно обработать только первый из них и отправить NAck (команда *LinkADRAns* со всеми битами состояния, установленными в 0) в ответ на все остальные ADR командные блоки. Устройство должно обрабатывать поля «Скорость передачи данных» (DataRate), «Выходная мощность передатчика» (TXPower) и «Число передач» (NbTrans) только из последней команды *LinkAdrReq* в последовательности ADR командного блока, так как значения этих параметров определяют общее состояние устройства. В поле «Подтверждение получения сообщения» (ACK) бит маски канала в ответе должен отражать принятие/отклонение окончательного плана каналов после обработки всех элементов управления маской канала в последовательности ADR командного блока.

Частота каждого канала задается для конкретного региона и определена в разделе 9. Оконечное устройство отвечает на команду *LinkADRReq* командой *LinkADRAns*. Структура команды *LinkADRAns* приведена на рисунке 21.

Размер (в байтах)	1
Данные LinkADRAns (LinkADRAns Payload)	Статус (Status)

Рисунок 21 — Структура команды *LinkADRAns*

Структура поля «Статус» (Status) команды *LinkADRAns* приведена на рисунке 22.

Биты	[7:3]	2	1	0
Статус (Status)	RFU	Выходная мощность передатчика в поле «Подтверждение получения сообщения» (Power ACK)	Запрошенная скорость передачи в поле «Подтверждение получения сообщения» (Data rate ACK)	Маска канала в поле «Подтверждение получения сообщения» (Channel mask ACK)

Рисунок 22 — Структура поля «Статус» (Status) команды *LinkADRAns*

Биты поля «Статус» (Status) команды *LinkADRAns* имеют значения согласно таблице 8.

Таблица 8 — Кодирование поля «Статус» (Status) команды *LinkADRAns*

Атрибут	Бит = 0	Бит = 1
Маска канала в поле «Подтверждение получения сообщения» (Channel mask ACK)	Отправленная маска канала разрешает использование пока неопределенного канала или маска канала требует отключения всех каналов. Команда была отклонена, и состояние оконечного устройства не было изменено	Отправленная маска канала успешно интерпретирована. Статусы всех в настоящее время определенных каналов были установлены в соответствии с маской
Запрошенная скорость передачи данных в поле «Подтверждение получения сообщения» (Data rate ACK)	Запрошенная скорость передачи данных неизвестна оконечному устройству или невозможно обеспечить заданную маску канала (не поддерживается ни одним из включенных каналов). Команда была отклонена, и состояние оконечного устройства не было изменено	Скорость передачи данных была успешно установлена или поле «Запрошенная скорость передачи» (DataRate) в запросе было установлено в значение 15, и он был проигнорирован
Выходная мощность передатчика в поле «Подтверждение получения сообщения» (Power ACK)	Устройство не может работать на уровне или ниже заданного уровня выходной мощности передатчика. Команда была отклонена, и состояние оконечного устройства не было изменено	Устройство может работать на уровне или ниже заданного уровня выходной мощности передатчика, или поле «Выходная мощность передатчика» (TXPower) в запросе было установлено в значение 15, и он был проигнорирован

Если любой из этих трех битов равен 0, то это значит, что команда не выполнена и устройство сохранило прежнее состояние.

6.3.4 Рабочий цикл устройства (*DutyCycleReq, DutyCycleAns*)

Команда *DutyCycleReq* используется сервером сети, чтобы ограничить оконечному устройству время на передачу сообщений в радиоэфире. Совокупное ограничение для всех частот соответствует ограничению для каждой используемой частоты. Структура команды *DutyCycleReq* приведена на рисунке 23.

Размер (в байтах)	1
Данные DutyCycleReq (DutyCycleReq Payload)	Рабочий цикл передачи (DutyCyclePL)

Рисунок 23 — Структура команды *DutyCycleReq*

Структура поля «Рабочий цикл передачи» (DutyCyclePL) команды *DutyCycleReq* приведена на рисунке 24.

Биты	[7:4]	[3:0]
Рабочий цикл передачи (DutyCyclePL)	RFU	Максимально допустимый рабочий цикл передачи (MaxDutyCycle)

Рисунок 24 — Структура поля «Рабочий цикл передачи» (DutyCyclePL) команды *DutyCycleReq*

Максимально допустимый рабочий цикл передачи вычисляется:

$$\textit{aggregated duty cycle} = \frac{1}{2\ \mathrm{MaxDutyCycle}}$$

Допустимый диапазон для MaxDutyCycle от 0 до 15. Значение 0 соответствует «нет ограничений», если в региональных настройках не указано иначе.

Оконечное устройство отвечает на *DutyCycleReq* командой *DutyCycleAns*. MAC-команда *DutyCycleAns* не содержит никаких полезных данных.

6.3.5 Параметры окон приема (*RXParamSetupReq, RXParamSetupAns*)

Команда *RXParamSetupReq* позволяет изменять частоту и скорость передачи данных, установленные для второго окна приема (RX2) после каждого восходящего сообщения. Эта команда также позволяет запрограммировать смещение скорости передачи данных в восходящей линии связи относительно скорости передачи данных в нисходящей линии связи в окно приема RX1. Структура команды *RXParamSetupReq* приведена на рисунке 25.

Размер (в байтах)	1	3
Данные RXParamSetupReq (RXParamSetupReq Payload)	Параметры нисходящей линии связи (DLsettings)	Частота (Frequency)

Рисунок 25 — Структура команды *RXParamSetupReq*

Структура поля «Параметры нисходящей линии связи» (DLsettings) команды *RXParamSetupReq* приведена на рисунке 26.

Биты	7	[6:4]	[3:0]
Параметры нисходящей линии связи (DLsettings)	RFU	RX1DRoffset	RX2DataRate

Рисунок 26 — Структура поля «Параметры нисходящей линии связи» (DLsettings) команды *RXParamSetupReq*

Поле RX1DRoffset задает смещение между скоростью передачи данных восходящей линии связи и скоростью передачи данных нисходящей линии связи, использованной для связи с оконечным устройством в первом окне приема (RX1). По умолчанию это смещение равно 0. Смещение используется, чтобы учитывать максимальные ограничения плотности покрытия базовых станций в некоторых регионах и балансировать загруженностью восходящей и нисходящей линий радиосвязи. Подробнее о значениях RX1DRoffset описано в разделе 9.

Поле RX2DataRate определяет скорость передачи данных нисходящей линии связи, используемой во втором окне приема RX2, следуя правилам, аналогичным для команды *LinkADRReq* (0 означает DR0/125кГц, к примеру).

Поле «Частота» (Frequency) соответствует частоте, используемой для второго окна приема RX2, при этом частота кодируется аналогично описанию команды *NewChannelReq*.

Команда *RXParamSetupAns* используется оконечным устройством, чтобы подтвердить прием команды *RXParamSetupReq*. Команда *RXParamSetupAns* должна добавляться в поле «Параметры кадра» (FOpts) всех восходящих сообщений, пока оконечное устройство не получит нисходящее сообщение в окно RX1 или RX2 (с длительностью RX2, характерной для устройства класса A). Это гарантирует, что даже при наличии потери восходящих пакетов сеть всегда в курсе параметров нисходящей линии связи, используемых оконечным устройством.

Структура команды *RXParamSetupAns* приведена на рисунке 27.

Размер (в байтах)	1
Данные RXParamSetupAns (RXParamSetupAns Payload)	Статус (Status)

Рисунок 27 — Структура команды *RXParamSetupAns*

Структура поля «Статус» (Status) команды *RXParamSetupAns* приведена на рисунке 28. Поле «Статус» (Status) команды *RXParamSetupAns* кодируется в соответствии с таблицей 9.

Биты	[7:3]	2	1	0
Статус (Status)	RFU	RX1DRoffset ACK	RX2 Data rate ACK	Channel ACK

Рисунок 28 — Структура поля «Статус» (Status) команды *RXParamSetupAns*

Таблица 9 — Кодирование поля «Статус» (Status) команды *RXParamSetupAns*

Атрибут	Бит = 0	Бит = 1
Channel ACK	Запрашиваемая частота не применима для оконечного устройства	Канал для окна приема RX2 успешно установлен
RX2 Data rate ACK	Запрошенная скорость передачи данных неизвестна оконечному устройству	Скорость передачи для окна приема RX2 успешно установлена
RX1DRoffset ACK	Смещение между скоростью передачи данных восходящей линии связи и скоростью передачи данных нисходящей линии связи для окна приема RX1 за пределами разрешенного диапазона	Смещение RX1DRoffset успешно установлено

Если любой из этих трех битов равен 0, то команда не выполнена, и устройство сохраняет прежнее состояние.

6.3.6 Статус оконечного устройства (*DevStatusReq, DevStatusAns*)

С помощью команды *DevStatusReq* сетевой сервер может запросить информацию о состоянии оконечного устройства. Команда не имеет атрибутов. Если оконечное устройство получило *DevStatusReq,* оно должно ответить командой *DevStatusAns*. Структура команды *DevStatusAns* приведена на рисунке 29.

Размер (в байтах)	1	1
Данные DevStatusAns (DevStatusAns Payload)	Уровень заряда батареи (Battery)	Статус (Status)

Рисунок 29 — Структура команды *DevStatusAns*

Поле «Уровень заряда батареи» (Battery) команды *DevStatusAns* кодируется в соответствии с таблицей 10.

Таблица 10 — Кодирование поля «Уровень заряда батареи» (Battery) команды *DevStatusAns*

Уровень заряда батареи	Описание
0	Оконечное устройство подключено к внешнему источнику питания
От 1 до 254	Уровень заряда батареи: 1 — находится на минимуме; 254 — находится на максимуме
255	Оконечное устройство не смогло измерить уровень заряда батареи

Поле «Устойчивость демодуляции» (Margin) содержит соотношение сигнал/шум (в дБ), измеренное при приеме последней команды — *DevStatusReq*. Значение поля «Устойчивость демодуляции» (Margin) округляется до ближайшего целого значения. Это целое 6-битовое число со знаком с минимальным значением минус 32 дБ и максимальным значением плюс 31 дБ.

Структура поля «Статус» (Status) команды *DevStatusAns* приведена на рисунке 30.

Биты	[7:6]	[5:0]
Статус (Status)	RFU	Устойчивость демодуляции (Margin)

Рисунок 30 — Структура поля «Статус» (Status) команды *DevStatusAns*

6.3.7 Создание/модификация канала (*NewChannelReq, NewChannelAns, DlChannelReq, DlChannelAns*)

Устройства, работающие в регионе, для которого определен фиксированный частотный план каналов, не должны выполнять эти MAC-команды (т. е. устройство не должно отвечать на команды). За-

даваемые каналы должны соответствовать требованиям региональных параметров, описанных в разделе 9.

Команда *NewChannelReq* может использоваться для изменения параметров существующего двунаправленного канала или создания нового. Команда задает центральную частоту нового канала и диапазон скоростей передачи данных в восходящей линии, которые могут использоваться на этом канале (рисунок 31).

Размер (в байтах)	1	3	1
Данные NewChannelReq (NewChannelReq Payload)	Индекс каналов (ChIndex)	Частота (Freq)	Диапазон скоростей передачи данных (DrRange)

Рисунок 31 — Структура команды *NewChannelReq*

Поле «Индекс каналов» (ChIndex) содержит индекс вновь созданного или измененного канала. Для каждого региона (см. раздел 9) устанавливаются каналы «по умолчанию», которые не могут быть изменены с помощью команды *NewChannelReq*.

Если число каналов «по умолчанию» равно *N*, то нумероваться каналы «по умолчанию» будут от 0 до [*N*-1], а от *N* до 15 будут нумероваться редактируемые каналы. Таким образом, Поле «Индекс каналов» (ChIndex) может принимать значение от *N* до 15. Устройство должно быть в состоянии обрабатывать по меньшей мере 16 различных каналов. В определенном регионе устройство может хранить параметры более 16 каналов.

Поле «Частота» (Freq) представляет собой 24-битовое целое число без знака. Фактическая частота канала в Гц считается 100 · Freq, где значения, представляющие частоты ниже 100 МГц, зарезервированы для использования в будущем. Это позволяет устанавливать частоту канала в диапазоне от 100 МГц до 1,67 ГГц с шагом 100 Гц. Значение Freq, равное 0, отключает канал. Оконечное устройство должно проверить, что частота на самом деле разрешена (обеспечивается) на аппаратном уровне радиомодуля, и вернуть сообщение об ошибке в противном случае.

Поле «Диапазон скоростей передачи данных» (DrRange) определяет диапазон скоростей передачи данных для восходящей линии связи, разрешенный для данного канала. Поле «Диапазон скоростей передачи данных» (DrRange) разделено на два 4-битных индекса (см. рисунок 32).

Биты	[7:4]	[3:0]
Диапазон скоростей передачи данных (DrRange)	Максимальная скорость передачи данных (MaxDR)	Минимальная скорость передачи данных (MinDR)

Рисунок 32 — Структура поля «Диапазон скоростей передачи данных» (DrRange) команды *NewChannelReq*

В соответствии с соглашениями, определенными в 6.3.3, поле «Минимальная скорость передачи данных» (MinDR) обозначает самый низкий уровень скорости передачи данных для восходящей линии связи, допустимый на этом канале. Например, используя региональные параметры Российской Федерации, 0 обозначает DR0 в полосе 125 кГц. Аналогичным образом поле «Максимальная скорость передачи данных» (MaxDR) обозначает самый высокий уровень скорости передачи данных для восходящей линии связи, допустимый на этом канале. Например, DrRange = 0x77 означает, что только 50 кбит/с GFSK допускается на канале и DrRange = 0x50 означает, что поддерживаются скорости передачи данных от DR0 в полосе 125 кГц до DR5 в полосе 125 кГц.

Измененный канал включается и сразу может быть использован для взаимодействия с сервером.

В окне приема RX1 частота в нисходящей линии устанавливается равной частоте в восходящей линии связи.

Оконечное устройство подтверждает получение *NewChannelReq* отправкой в ответ команды *NewChannelAns*. Структура команды *NewChannelAns* приведена на рисунке 33.

Размер (в байтах)	1
Данные NewChannelAns (NewChannelAns Payload)	Статус (Status)

Рисунок 33 — Структура команды *NewChannelAns*

Структура поля «Статус» (Status) команды *NewChannelAns* приведена на рисунке 34. Поле «Статус» (Status) команды *NewChannelAns* кодируется в соответствии с таблицей 9.

Биты	[7:2]	1	0
Статус (Status)	RFU	Data rate range ok	Channel frequency ok

Рисунок 34 — Структура поля «Статус» (Status) команды *NewChannelAns*

Таблица 11 — Кодирование поля «Статус» (Status) команды *NewChannelAns*

Атрибут	Бит = 0	Бит = 1
Data rate range ok	Указанный диапазон скоростей передачи данных превышает поддерживаемый данным оконечным устройством	Диапазон скоростей передачи данных совместим с возможностями оконечного устройства
Channel frequency ok	Устройство не может использовать данную частоту	Устройство имеет возможность использовать эту частоту

Если любой из этих двух битов равен 0, то команда не выполнена и новый канал не создан.

Команда *DlChannelReq* позволяет сети ассоциировать различные частоты нисходящей линии связи с окном приема RX1. Эта команда применяется для всех спецификаций физического уровня, поддерживающих команду *NewChannelReq* (поддерживается для региональных параметров Российской Федерации, Европы (EU) и Китая и не поддерживатся для США и Австралии).

Команда задает центральную частоту, используемую для нисходящей линии связи в окне приема RX1. Структура команды *DlChannelReq* приведена на рисунке 35.

Размер (в байтах)	1	3
Данные DlChannelReq (DlChannelReq Payload)	Индекс канала (ChIndex)	Частота (Freq)

Рисунок 35 — Структура команды *DlChannelReq*

Поле «Индекс канала» (ChIndex) — индекс канала, для которого изменяется частота нисходящей линии связи.

Поле «Частота» (Freq) представляет собой 24-битовое целое число без знака. Фактическая частота канала в Гц считается 100 · Freq, где значения, представляющие частоты ниже 100 МГц, зарезервированы для использования в будущем. Это позволяет устанавливать частоту канала в диапазоне от 100 МГц до 1,67 ГГц с шагом 100 Гц. Значение Freq, равное 0, отключает канал. Оконечное устройство должно проверить, что частота на самом деле поддерживается на аппаратном уровне радиомодулем, и вернуть сообщение об ошибке в противном случае.

Оконечное устройство подтверждает получение *DlChannelReq* отправкой в ответ команды *DlChannelAns*. Команда *DlChannelAns* должна добавляться в поле «Параметры кадра» (FOpts) всех восходящих сообщений, пока оконечное устройство не получит нисходящий пакет. Это гарантирует, что даже при наличии потери пакетов в восходящей линии связи сеть всегда будет в курсе частот, используемых оконечным устройством в нисходящей линии связи. Структура команды *DlChannelAns* приведена на рисунке 36.

Размер (в байтах)	1
Данные DlChannelAns (DlChannelAns Payload)	Статус (Status)

Рисунок 36 — Структура команды *DlChannelAns*

Структура поля «Статус» (Status) команды *DlChannelAns* приведена на рисунке 37. Поле «Статус» (Status) команды *DlChannelAns* кодируется в соответствии с таблицей 12.

Биты	[7:2]	1	0
Status	RFU	Uplink frequency exists	Channel frequency ok

Рисунок 37 — Биты поля «Статус» (Status) команды *DlChannelAns*

Таблица 12 — Кодирование поля «Статус» (Status) команды *DlChannelAns*

Атрибут	Бит = 0	Бит = 1
Channel frequency ok	Устройство не может использовать эту частоту	Устройство имеет возможность использовать эту частоту
Uplink frequency exists	Частота восходящей линии связи не определена для данного канала, частота нисходящей линии связи может быть установлена только для канала, который уже имеет допустимую частоту восходящей линии связи	Частота восходящей линии связи для канала допустима (корректна)

6.3.8 Настройка задержки между TX и RX (*RXTimingSetupReq, RXTimingSetupAns*)

Команда *RXTimingSetupReq* позволяет настраивать задержку между окончанием передачи восходящего сообщения (TX) и открытием первого окна приема (RX1). Второе окно приема (RX2) открывается через 1 (одну) секунду после первого окна приема. Структура команды *RXTimingSetupReq* приведена на рисунке 38.

Размер (в байтах)	1
RXTimingSetupReq Payload	Settings

Рисунок 38 — Структура команды *RXTimingSetupReq*

Поле Delay определяет время задержки. Поле разделено на два 4-разрядных индекса. Структура поля Delay команды *RXTimingSetupReq* приведена на рисунке 39. Поле Delay команды *RXTimingSetupReq* кодируется в соответствии с таблицей 13.

Биты	[7:4]	[3:0]
Settings	RFU	Del

Рисунок 39 — Структура поля Delay команды *RXTimingSetupReq*

Задержка (Del) — указывается в секундах. Значение Del, равное 0, соответствует 1 с.

Таблица 13 — Кодирование поля Delay команды *RXTimingSetupReq*

Del	Задержка (с)
0	1
1	1
2	2
3	3
…	…
15	15

Оконечное устройство отвечает на команду *RXTimingSetupReq* отправкой команды *RXTimingSetupAns* без атрибутов.

Команда *RXTimingSetupAns* должна добавляться в поле «Параметры кадра» (FOpts) всех восходящих сообщений, пока оконечное устройство не получит нисходящий пакет в окно RX1 или RX2 (с длительностью RX2, характерной для устройства класса A). Это гарантирует, что даже при наличии потери пакетов в восходящей линии связи сеть будет всегда в курсе параметров нисходящей линии связи, используемых оконечным устройством.

6.3.9 Параметры передачи оконечного устройства (*TxParamSetupReq, TxParamSetupAns*)

Эта MAC-команда должна выполняться с соблюдением региональных параметров (см. раздел 9).

Команда *TxParamSetupReq* может быть использована для уведомления оконечного устройства о максимально допустимом времени задержки, т. е. максимальном времени непрерывной передачи пакета по радиоэфиру, а также максимально допустимой эффективной изотропной мощности излучения (EIRP) оконечного устройства. Структура команды *TxParamSetupReq* приведена на рисунке 40.

Размер (в байтах)	1
Данные TxParamSetupReq (TxParamSetupReq Payload)	EIRP_DwellTime

Рисунок 40 — Структура команды *TxParamSetupReq*

Структура поля EIRP_DwellTime команды *TxParamSetupReq* приведена на рисунке 41.

Биты	[7:6]	5	4	[3:0]
EIRP_DwellTime	RFU	Максимальное время задержки в нисходящей линии связи (DownlinkDwellTime)	Максимальное время задержки в восходящей линии связи (UplinkDwellTime)	Максимальное значение мощности излучения (MaxEIRP)

Рисунок 41 — Структура поля EIRP_DwellTime команды *TxParamSetupReq*

Биты [0...3] команды *TxParamSetupReq* кодируют максимальное значение мощности излучения — MaxEIRP. Значения MaxEIRP охватывают широкий диапазон, включающий параметры всех возможных регионов. При установлении MaxEIRP следует соблюдать региональные параметры, приведенные в разделе 9.

MaxEIRP кодируется значениями согласно рисунку 42.

Кодируемое значение	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MaxEIRP (dBm)	8	10	12	13	14	16	18	20	21	24	26	27	29	30	33	36

Рисунок 42 — Кодирование MaxEIRP

Значение MaxEIRP соответствует верхней границе выходной мощности передатчика устройства. Устройство не обязано передавать на этой мощности, но никогда не должно превышать указанное значение.

Биты UplinkDwellTime и DownlinkDwellTime определяют максимальное время задержки в восходящей и нисходящей линиях связи соответственно, которые кодируются согласно таблице 14.

Таблица 14 — Кодирование значений задержки

Кодируемое значение	Время задержки
0	Не ограничено
1	400 мс

Если в соответствии с региональными параметрами данная MAC-команда поддерживается, то оконечное устройство отвечает на команду *TxParamSetupReq* отправкой команды *TxParamSetupAns*. Ответ *TxParamSetupAns* не содержит атрибутов.

Если региональные параметры не допускают использование данной MAC-команды, то устройство не обрабатывает ее и не передает подтверждения.

6.3.10 Оповещение об обновлении ключей (*RekeyInd, RekeyConf*)

Эта MAC-команда доступна только для OTA-устройств, активированных в сети с сервером сети, поддерживающим LoRaWAN v.1.1. Сервер сети, поддерживающий только LoRaWAN v.1.0, не реализует эту MAC-команду.

ABP-устройства не должны выполнять эту команду. Сетевой сервер должен игнорировать команду *RekeyInd*, поступившую от ABP-устройства.

Для OTA-устройств MAC-команда *RekeyInd* используется для подтверждения обновления ключей безопасности и в будущих версиях LoRaWAN RU (>1.1), чтобы договориться об используемой между оконечным устройством и сервером сети версии протокола LoRaWAN RU. Команда не является оповещением о сбросе (перезагрузке) параметров MAC и радиоканала (см. 6.4.2.3).

Команда *RekeyInd* включает в себя номер младшей версии LoRaWAN RU. Структура команды *RekeyInd* приведена на рисунке 43.

Размер (в байтах)	1
Данные RekeyInd (RekeyInd Payload)	Версия LoRaWAN RU устройства (Dev LoRaWAN RU version)

Рисунок 43 — Структура команды *RekeyInd*

Структура поля «Версия LoRaWAN RU устройства» (Dev LoRaWAN RU version) команды *RekeyInd* приведена на рисунке 44.

Биты	[7:4]	[3:0]
Версия LoRaWAN RU устройства (Dev LoRaWAN RU version)	RFU	Дополнительный номер версии (Minor)=1

Рисунок 44 — Структура поля «Версия LoRaWAN RU устройства» (Dev LoRaWAN RU version) команды *RekeyInd*

Поле «Дополнительный номер версии» (Minor) указывает на дополнительный номер версии LoRaWAN RU, поддерживаемой оконечным устройством (см. таблицу 15).

Таблица 15 — Значения поля «Дополнительный номер версии» (Minor)

Дополнительный номер версии (Minor)	Значение поля
RFU	0
1 (LoRaWAN RU x.1)	1
RFU	От 2 до 15

OTA-устройства должны отправлять команду *RekeyInd* во всех восходящих сообщениях, требующих и не требующих подтверждения, после успешной обработки Join-Accept (новые сеансовые ключи были получены), пока не будет получена команда *RekeyConf*. Если устройство не получило *RekeyConf* в течение первых ADR_ACK_LIMIT восходящих сообщений, то оно должно вернуться в состояние присоединения к сети. Команды *RekeyInd*, присланные от этих устройств после этого, должны быть проигнорированы сервером сети. Сервер сети должен отклонить все восходящие кадры, защищенные новыми сеансовыми ключами, которые были получены после отправки Join-Accept и до первого восходящего кадра, который несет в себе команду *RekeyInd*.

Когда сетевой сервер получает *RekeyInd*, он отвечает командой *RekeyConf*.

Команда *RekeyConf* содержит один байт полезных данных с закодированной версией LoRaWAN RU, поддерживаемой сервером сети, используется тот же формат, что и для «Dev LoRaWAN RU version». Структура команды *RekeyConf* приведена на рисунке 45.

Размер (в байтах)	1
Данные RekeyConf (RekeyConf Payload)	Версия LoRaWAN RU сервера сети (Serv LoRaWAN RU version)

Рисунок 45 — Структура команды *RekeyConf*

Версия протокола сервера должна быть больше, чем 0 (0 не допускается), и меньше либо равна версии протокола устройства. Поэтому для устройства, поддерживающего настоящий стандарт в объеме спецификации LoRaWAN 1.1, единственным допустимым значением является 1. Если версия сервера недействительна, устройство должно отбросить команду *RekeyConf* и повторно отправить (ретранслировать) команду *RekeyInd* в следующем восходящем кадре.

6.3.11 Параметры ADR (*ADRParamSetupReq, ADRParamSetupAns*)

Команда *ADRParamSetupReq* позволяет изменять параметры ADR_ACK_LIMIT и ADR_ACK_DELAY, определенные в алгоритме обратного переключения ADR.

Структура команды *ADRParamSetupReq* приведена на рисунке 46.

Размер (в байтах)	1
Данные ADRParamSetupReq (ADRParamSetupReq Payload)	ADRparam

Рисунок 46 — Структура команды *ADRParamSetupReq*

Структура ADRparam команды *ADRParamSetupReq* приведена на рисунке 47.

Биты	[7:4]	[3:0]
ADRparam	Limit_exp	Delay_exp

Рисунок 47 — Структура ADRparam команды *ADRParamSetupReq*

Поле Limit_exp устанавливает значение параметра ADR_ACK_LIMIT.

$$ADR_ACK_LIMIT = 2^{Limit_exp}$$

Допустимый диапазон значений Limit_exp находится в пределах от 0 до 15, что соответствует диапазону значений от 1 до 32768 для параметра ADR_ACK_LIMIT.

Поле Delay_exp устанавливает значение параметра ADR_ACK_DELAY.

$$ADR_ACK_DELAY = 2^{Delay_exp}$$

Допустимый диапазон значений Delay_exp находится в пределах от 0 до 15, что соответствует диапазону значений от 1 до 32768 для параметра ADR_ACK_DELAY.

Команда *ADRParamSetupAns* используется оконечным устройством, чтобы подтвердить прием команды *ADRParamSetupReq*. Команда *ADRParamSetupAns* не имеет поля данных.

6.3.12 Время устройства (*DeviceTimeReq, DeviceTimeAns*)

Данная MAC-команда доступна, только если устройство активировано в сети с сервером сети, поддерживающим LoRaWAN 1.1. Сервер сети, поддерживающий только LoRaWAN 1.0, не реализует эту MAC-команду.

С помощью команды *DeviceTimeReq* оконечное устройство может запрашивать у сети текущие дату и время сети. Запрос не имеет никаких полезных данных.

С помощью команды *DeviceTimeAns* сетевой сервер предоставляет оконечному устройству дату и время сети. Предоставленное время — это время сети, зафиксированное в конце передачи восходящего сообщения. Команда *DeviceTimeAns* имеет 5 байт полезных данных. Структура команды *DeviceTimeAns* приведена на рисунке 48.

Размер (в байтах)	4	1
Данные DeviceTimeAns (DeviceTimeAns Payload)	32-битное целое без знака: число секунд с начала эпохи	8-битное целое без знака: доли секунды с шагом в $1/2^8$ секунды

Рисунок 48 — Структура команды *DeviceTimeAns*

Время, предоставленное сетью, должно иметь точность не хуже +/−100 мС.

Примечание — В качестве начальной точки отсчета времени эпохи используется 6 января 1980 года, полночь. Поле «секунды» — это количество секунд, прошедшее с момента начала эпохи. Это поле монотонно увеличивается каждую секунду на 1. Чтобы преобразовать это поле в UTC время, високосные секунды должны быть приняты во внимание.

Пример — пятница, 12 февраля 2016 года, 14:24:31 по Гринвичу соответствует 1139322288 секунд с начала эпохи по шкале GPS. По состоянию на июнь 2017 года время GPS на 17 секунд впереди времени UTC.

6.3.13 Вынужденное повторное присоединение к сети (*ForceRejoinReq*)

С помощью команды *ForceRejoinReq* сеть запрашивает у устройства немедленно передать сообщение с запросом повторного присоединения 0-го или 2-го типа (Rejoin-Request type 0 or type 2) с установленным числом и периодичностью попыток присоединения и скоростью передачи данных. Этот восходящий *RejoinRequest* может быть использован сетью для немедленной замены ключей устройства или для инициирования процедуры передачи устройства в другой сервер сети в роуминге.

Команда *ForceRejoinReq* имеет два байта полезных данных. Структура команды *ForceRejoinReq* приведена на рисунке 49.

Биты	[15:14]	[13:11]	[10:8]	7	[6:4]	[3:0]
ForceRejoinReq	RFU	Period	Max_Retries	RFU	RejoinType	DR

Рисунок 49 — Структура команды *ForceRejoinReq*

Параметры кодируются следующим образом.

Period — задержка между повторами передачи, должна быть равна:

$$\text{Period} = 32 \cdot 2^{\text{Period}} + \text{Rand32},$$

где Rand32 — это псевдослучайное число в диапазоне [0..32].

Max_Retries — общее количество попыток, которые выполнит устройство, чтобы отправить запрос на повторное присоединение к сети (Rejoin-Request).

- 0: запрос на повторное присоединение к сети будет отправлен только один раз (без повтора).
- 1: запрос на повторное присоединение к сети должен быть отправлен два раза в общей сложности (1 + 1 повтор).
- ...
- 7: запрос на повторное присоединение к сети должен быть отправлен восемь раз (1 + 7 повторов).

Поле RejoinType указывает тип запроса *RejoinRequest*, который будет передан устройством:

- 0 или 1: должен быть передан запрос *RejoinRequest* типа 0.
- 2: должен быть передан запрос *RejoinRequest* типа 2.
- ... 7: зарезервированы для последующего использования.

DR-кадр с RejoinRequest должен быть передан с указанной скоростью передачи данных. Соотношение между реальной физической скоростью передачи данных (обусловленной типом используемой модуляции) и значением скорости передачи данных определяется теми же правилами, что и для команды LinkADRReq, и определяется в соответствии с региональными параметрами (см. раздел 9).

Команда не имеет ответа, так как устройство должно отправить *RejoinRequest* при получении команды. Первая передача сообщения с *RejoinRequest* должна быть осуществлена сразу же после приема команды (но сеть может не получить его). Если устройство получает новую команду *ForceRejoinReq* прежде, чем оно достигнет максимального числа повторных передач, то устройство должно продолжить передачу *RejoinRequest* с новыми параметрами.

6.3.14 Параметры повторного присоединения к сети (*RejoinParamSetupReq*, *RejoinParamSetupAns*)

С помощью команды *RejoinParamSetupReq* сеть может запросить устройство периодически отправлять сообщение с запросом на повторное присоединение к сети типа 0 (RejoinRequest type 0) с заданной периодичностью отправки, определенной как время или число восходящих кадров.

И время, и число кадров предлагаются для использования устройствами, которые могут не иметь возможности измерять время. Заданная периодичность устанавливает максимальное время и количество восходящих кадров между двумя отправками *RejoinRequest*. Устройство может передавать *RejoinRequest* чаще заданной периодичности. Команда имеет единственный байт полезных данных. Структура команды *RejoinParamSetupReq* приведена на рисунке 50.

Биты	[7:4]	[3:0]
RejoinParamSetupReq	MaxTimeN	MaxCountN

Рисунок 50 — Структура команды *RejoinParamSetupReq*

Параметры определены следующим образом:

MaxCountN = C = от 0 до 15.

Устройство должно отправлять *RejoinRequest* типа 0 не реже, чем каждое 2C+4 исходящее сообщение.

MaxTimeN = T = от 0 до 15.

Устройство должно отправлять *RejoinRequest* типа 0 не реже, чем каждые 2T+10 секунд.

- T = 0 соответствует примерно 17 минутам.
- T = 15 — около одного года.

Устройство должно обеспечивать повторное присоединение к сети по достижении порогового значения количества восходящих кадров. Периодичность повторного присоединения, основанная на времени, является необязательной. Устройство, которое не может реализовать подсчет временного интервала, должно известить об этом в ответе. Ответ содержит один байт полезных данных. Структура команды *RejoinParamSetupReq* приведена на рисунке 51.

Биты	[7:1]	0
RejoinParamSetupReq	RFU	TimeOK

Рисунок 51 — Структура команды *RejoinParamSetupReq*

Если бит 0 равен 1, то прибор принял установку периодичности в формате времени и количества восходящих кадров, в противном случае он принимает установку периодичности только в виде ограничения количества восходящих кадров.

6.4 Активация оконечного устройства

Для работы в сетях LoRaWAN RU каждое оконечное устройство должно быть зарегистрировано и активировано.

Активация оконечного устройства может быть выполнена двумя способами: по воздуху (OTAA) или через персонализацию (ABP).

6.4.1 Данные об активации, сохраняемые в устройстве

6.4.1.1 Перед активацией

а) JoinEUI

JoinEUI является глобальным идентификатором приложения в [1] EUI64 адресного пространства, который однозначно идентифицируется сервером присоединения к сети (Join-Server), который обеспечивает выполнение процедуры присоединения к сети и производства сеансовых ключей.

Для устройств, поддерживающих активацию по воздуху, JoinEUI должен быть сохранен в энергонезависимой памяти устройства до начала выполнения процедуры присоединения.

Для устройств, поддерживающих только активацию через персонализацию, наличие JoinEUI не требуется.

б) DevEUI

DevEUI — это глобальный идентификатор оконечного устройства в [1] EUI64 адресном пространстве, который однозначно идентифицирует оконечное устройство.

DevEUI рекомендуется использовать сетевым серверам в качестве уникального идентификатора устройства, независимо от используемого метода активации устройства, для идентификации устройства при межсетевом роуминге (перемещении устройства из одного сегмента сети в другой).

Для устройств, поддерживающих активацию по воздуху, DevEUI должен быть сохранен в энергонезависимой памяти устройства до начала выполнения процедуры присоединения.

Для устройств, поддерживающих только активацию через персонализацию, DevEUI не требуется сохранять в энергонезависимой памяти устройства, но рекомендуется это сделать.

Примечание — Рекомендуется использовать DevEUI также в качестве этикетки устройства (номенклатурного номера) и для управления устройством (учета и сопровождения).

в) Первичные ключи устройства (AppKey и NwkKey)

Ключи NwkKey и AppKey являются первичными ключами, индивидуальными для одного экземпляра оконечного устройства. Ключи NwkKey и AppKey назначаются оконечному устройству при его производстве. В ходе процедуры присоединения оконечного устройства к сети методом активации по воздуху NwkKey используется для получения сеансовых ключей FNwkSIntKey, SNwkSIntKey, NwkSEncKey (LoRaWAN v.1.1) и NwkSkey (LoRaWAN v.1.0); и AppKey используется для получения сеансового ключа AppSKey.

Примечание — При работе с сетевым сервером LoRaWAN v1.1 для получения сеансового ключа приложения используется только AppKey, поэтому NwkKey может быть использован в сети для управления процедурой присоединения без возможности оператора сети раскодировать данные приложений.

Требования к системе безопасности сервера сети выходят за рамки настоящего стандарта и должны рассматриваться отдельно.

Для обеспечения обратной совместимости с LoRaWAN v.1.0 и более ранних версий сетевых серверов, которые не поддерживают два первичных ключа, оконечные устройства по умолчанию должны быть настроены на использование при присоединении к сети схемы с одним первичным ключом. Признаком такого состояния оконечного устройства является установленный в ноль бит № 7 «OptNeg» в поле DLsetting в сообщении Join-Accept.

Устройство в этом случае должно:

- использовать NwkKey для получения сеансовых ключей AppSKey и FNwkSIntKey, как указано в спецификации LoRaWAN v.1.0;
- установить SNwkSIntKey и NwkSEncKey равными FNwkSIntKey. Один и тот же сетевой сеансовый ключ используется для расчета MIC восходящих и нисходящих сообщений и кодирования кадров данных (MACPayload) в соответствии со спецификацией LoRaWAN v.1.0.

NwkKey должен быть сохранен в энергонезависимой памяти устройства, поддерживающего использование процедуры OTAA.

NwkKey не требуется оконечным устройствам, поддерживающим только процедуру ABP.

AppKey должен быть сохранен в энергонезависимой памяти устройства, поддерживающего использование процедуры OTAA.

AppKey не требуется оконечным устройствам, поддерживающим только процедуру ABP.

NwkKey и AppKey должны храниться таким образом, чтобы исключить возможность их извлечения и повторного использования злоумышленниками.

Примечание — Поскольку все оконечные устройства оснащены уникальными для каждого оконечного устройства первичными ключами AppKey и NwkKey, то в случае извлечения AppKey/NwkKey из оконечного устройства компрометируется только это одно оконечное устройство.

г) Формирование JSIntKey и JSEncKey

Для OTA-устройств из первичного ключа NwkKey формируются два специальных ключа на весь жизненный цикл устройства:

- JSIntKey используется для формирования MIC в сообщении с запросом Rejoin-Request первого типа и ответа Join-Accept;
- JSEncKey используется для кодирования Join-Accept, сформированного в ответ на запрос Rejoin-Request.

Рекомендуется использовать алгоритм блочного шифрования. Описание и выбор конкретных алгоритмов шифрования данных выходят за рамки настоящего стандарта.

Для вычисления JSIntKey с помощью ключа NwkKey кодированию подлежит сообщение:

$$message = 0x06 \,||\, DevEUI \,||\, Pad_{16}$$

Для вычисления JSEncKey с помощью ключа NwkKey кодированию подлежит сообщение:

$$message = 0x05 \,||\, DevEUI \,||\, Pad_{16}$$

6.4.1.2 После активации

После активации в оконечном устройстве хранятся следующие дополнительные сведения:

- короткий адрес оконечного устройства (DevAddr);
- три сетевых сеансовых ключа (NwkSEncKey/SNwkSIntKey/FNwkSIntKey);
- сеансовый ключ приложения (AppSKey).

а) Короткий адрес оконечного устройства (DevAddr)

DevAddr состоит из 32 битов, идентифицирующих оконечное устройство в текущей (существующей) сети. Структура DevAddr приведена на рисунке 52.

Биты	От 31 до 32-*N*	От 31-*N* до 0
DevAddr	AddrPrefix	NwkAddr

Рисунок 52 — Структура DevAddr

N — целое число в диапазоне [7:24].

Протокол LoRaWAN RU поддерживает различные типы сетевых адресов с разным размером адресного пространства. Переменный размер поля AddrPrefix является производным от уникального идентификатора сервера сети NetID (см. 6.4.2.3), за исключением значений AddrPrefix, зарезервированных для экспериментальных/частных сетей. Поле AddrPrefix позволяет сетевым серверам обнаруживать и в реальном времени управлять устройствами в роуминге. Устройства, которые не соблюдают это правило, не смогут переподключаться между двумя сетями, т. к. будет невозможно найти их домашний сетевой сервер.

Младшие (32-*N*) биты DevAddr — это сетевой адрес оконечного устройства (NwkAddr), который может назначаться по усмотрению администратора сети.

Значения AddrPrefix, указанные на рисунке 53, могут быть использованы любой частной/экспериментальной сетью и не будут взаимодействовать в роуминге.

AddrPrefix, зарезервированные для частных/экспериментальных сетей
N = 7
AddrPrefix = 7'b0000000 или AddrPrefix = 7'b0000001
NwkAddr — это 25 бит, назначаются по усмотрению администратора сети

Рисунок 53 — Значения AddrPrefix

б) Сетевой сеансовый ключ целостности восходящих сообщений (FNwkSIntKey)

FNwkSIntKey (forwarding network session integrity key) — это сетевой сеансовый ключ, определенный для оконечного устройства. Он используется оконечным устройством для вычисления MIC или части MIC всех восходящих сообщений с данными для обеспечения их целостности.

FNwkSIntKey должен храниться таким образом, чтобы предотвратить извлечение и повторное использование злоумышленниками.

в) Сетевой сеансовый ключ целостности нисходящих сообщений (SNwkSIntKey)

SNwkSIntKey (serving network session integrity key) — это сетевой сеансовый ключ, определенный для оконечного устройства. Он используется оконечным устройством для проверки MIC всех нисходящих сообщений с данными для обеспечения целостности данных и вычисления половины MIC восходящих сообщений.

Примечание — Для расчета MIC восходящих сообщений используется два ключа (FNwkSIntKey и SNwkSIntKey) для того, чтобы обеспечить переадресацию сетевого сервера в настройках роуминга и возможность проверять только половину поля MIC.

Когда устройство подключается к сети LoRaWAN 1.0, сетевой сервер использует один и тот же ключ для расчета MIC как восходящих, так и нисходящих сообщений, в этом случае SNwkSIntKey принимает значение, совпадающее с FNwkSIntKey.

SNwkSIntKey должен храниться таким образом, чтобы предотвратить извлечение и повторное использование злоумышленниками.

г) Сетевой сеансовый ключ кодирования (NwkSEncKey)

NwkSEncKey (network session encryption key) — это сетевой сеансовый ключ, определенный для оконечного устройства. Он используется для кодирования и раскодирования восходящих и нисходящих MAC-команд, передаваемых в поле прикладных данных FRMPayload на порт 0 или в поле FOpt. Когда устройство подключается к сети LoRaWAN 1.0, сервер использует один и тот же ключ для кодирования MAC-сообщения (MACPayload) и расчета MIC. В этом случае NwkSEncKey принимает значение, совпадающее с FNwkSIntKey.

NwkSEncKey должен храниться таким образом, чтобы предотвратить извлечение и повторное использование злоумышленниками.

д) Сеансовый ключ приложения (AppSKey)

AppSKey (application session key) — это сеансовый ключ приложения, определенный для оконечного устройства. Он используется сервером приложений и оконечным устройством для кодирования и декодирования поля прикладных данных (FRMPayload) в информационных сообщениях при условии значения поля FPort от 1 до 224. AppSKey должен храниться таким образом, чтобы предотвратить извлечение и повторное использование злоумышленниками.

е) Контекст сеанса связи

Контекст сеанса связи включает параметры сетевого сеанса и параметры сеанса приложения.

В рамках сетевого сеанса определяются следующие сущности:

- FNwkSIntKey/SNwkSIntKey — сетевые сеансовые ключи целостности;
- NwkSEncKey — сетевой сеансовый ключ кодирования;
- FCntUp — счетчик восходящих по сети кадров;
- FCntDown (LoRaWAN 1.0) и NFCntDown (LoRaWAN 1.1) — счетчики нисходящих по сети кадров;
- DevAddr — адрес оконечного устройства.

В рамках сеанса приложения определяются следующие сущности:

- AppSKey — сеансовый ключ приложения;
- FCntUp — счетчик восходящих кадров;
- FCntDown (LoRaWAN 1.0) и AFCntDown (LoRaWAN 1.1) — счетчики нисходящих кадров.

Состояние сетевого сеанса поддерживается сервером сети и оконечным устройством. Состояние сеанса приложения поддерживается сервером приложений и оконечным устройством.

По окончании процедуры авторизации OTAA или ABP устанавливается новый безопасный сеанс связи между сервером сети (или сервером приложений) и оконечным устройством. Ключи и адрес оконечного устройства являются фиксированными в течение всего срока сеанса связи (FNwkSIntKey, SNwkSIntKey, AppSKey, DevAddr). Счетчики кадров увеличиваются по факту обмена пакетами в течение сеанса связи (FCntUp, FCntDown, NFCntDown, AFCntDown).

Для устройств OTAA счетчики кадров не должны повторно использоваться при одних и тех же сеансовых ключах, поэтому новый сеанс должен быть установлен до момента переполнения счетчиков кадров. Таким образом, количество отправленных сообщений с одними и теми же сеансовыми ключами не может превышать 4 294 967 295 сообщений.

Рекомендуется, чтобы состояние сеанса сохранялось даже при переключении (выключении и включении) питания на оконечном устройстве. Невыполнение этого требования для устройств OTAA означает, что процедуру активации придется выполнять при каждом выключении и включении устройства.

6.4.2 Активация по воздуху (OTAA)

Для активации по воздуху оконечное устройство должно пройти процедуру присоединения к сети, прежде чем участвовать в обмене данными с сетевым сервером. Оконечное устройство должно проходить через новую процедуру присоединения к сети каждый раз при потере сведений о параметрах сеансного уровня связи.

Прежде чем начнется процедура присоединения к сети, необходимо, чтобы оконечное устройство было персонализировано следующей информацией: DevEUI, JoinEUI, NwkKey и AppKey.

Примечание — Для активации по воздуху устройства не персонализируются парой сетевых сеансовых ключей. Вместо этого всякий раз, когда оконечное устройство присоединяется к сети, для кодирования и проверки целостности передаваемых на сетевом уровне данных формируются сетевые сеансовые ключи, специфические для данного устройства. Таким образом, роуминг оконечных устройств между сетями различных провайдеров облегчается. Использование разных сетевых сеансовых ключей и сеансового ключа приложения позволяет дополнительно ограничить сетевые сервера, на которых данные приложений не должны быть прочитаны или изменены сетевым провайдером.

6.4.2.1 Процедура присоединения к сети

Со стороны оконечного устройства процедура присоединения к сети представляет собой отправку оконечным устройством запроса на присоединение к сети (Join-Request) или повторное присоединение к сети (Rejoin-Request) и получение подтверждения присоединения (переприсоединения) к сети (Join-Accept).

6.4.2.2 Сообщение с запросом на присоединение к сети (Join-Request)

Процедура присоединения всегда инициируется оконечным устройством путем отправки сообщения с запросом на присоединение. Структура сообщения приведена на рисунке 54.

Размер (в байтах)	8	8	2
Запрос на присоединение к сети	JoinEUI	DevEUI	DevNonce

Рисунок 54 — Структура сообщения с запросом на присоединение к сети

Сообщение с запросом на присоединение к сети содержит JoinEUI и DevEUI оконечного устройства с последующим полем случайного значения из 2 байтов (DevNonce).

DevNonce — идентификатор запросов на присоединение, должен быть случайным числом (для обратной совместимости со спецификациями LoRaWAN v.1.0.x) или счетчиком (для обратной совместимости со спецификацией LoRaWAN v.1.1).

Если DevNonce — это счетчик, то его значение начинается с 0, когда устройство изначально включается и увеличивается с каждым JoinRequest. Значение DevNonce никогда не должно использоваться повторно для заданного значения JoinEUI. Если оконечное устройство может быть выключено и включено, то DevNonce не должен изменяться (должен сохраняться в энергонезависимой памяти). Сброс DevNonce без изменения JoinEUI вызовет отклонение сетевым сервером запроса устройства на присоединение к сети. Для каждого оконечного устройства сетевой сервер отслеживает значения DevNonce, использованные оконечным устройством, и игнорирует запросы на присоединение к сети, если DevNonce не изменился (не увеличился). Когда счетчик DevNonce переполняется (предыдущее значение счетчика равно 16 777 215), эксплуатация оконечного устройства завершается.

Примечание — Этот механизм предотвращает атаки повторного воспроизведения путем отправки ранее записанных сообщений — запросов на присоединение с целью отключения соответствующего оконечного устройства от сети. Сетевой сервер в любое время обработает запрос на присоединение и сформирует пакет с подтверждением присоединения, он должен поддерживать и старые параметры контекста сеанса (ключи и счетчики, если они есть), и новые, пока не получит первый успешный пакет из восходящей линии связи, содержащий команду *RekeyInd* с использованием настроек нового сеанса. После этого настройки старого сеанса могут быть безопасно удалены.

Значение MIC для сообщения с запросом на присоединение рассчитывается с помощью первичного сеансового ключа оконечного устройства — NwkKey.

Рекомендуется использовать алгоритм блочного шифрования. Описание и выбор конкретных алгоритмов шифрования данных выходят за рамки настоящего стандарта.

Кодированию с помощью ключа первичного сеансового ключа оконечного устройства (NwkKey) подлежит сообщение:

message = MHDR || JoinEUI || DevEUI || DevNonce.

Сообщение с запросом на присоединение к сети не кодируется и может передаваться на любой скорости передачи данных и частоте, случайным образом выбранной из возможных для присоединения каналов. Рекомендуется использовать различные скорости передачи данных. Интервалы между передачами запросов на присоединение должны соблюдать условия, описанные в 6.5. Для каждой следующей передачи запроса на подключение устройство должно увеличить значение DevNonce.

6.4.2.3 Сообщение с подтверждением присоединения (переприсоединения) к сети (Join-Accept)

Сетевой сервер отвечает на сообщение с запросом на присоединение (или повторное присоединение) сообщением с подтверждением присоединения (переприсоединения) к сети, если оконечному устройству разрешено присоединение к сети. Сообщение с подтверждением соединения отправляется как обычное нисходящее сообщение, но использует задержки JOIN_ACCEPT_DELAY1 или JOIN_ACCEPT_DELAY2 (вместо RECEIVE_DELAY1 и RECEIVE_DELAY2 соответственно). Частота канала и

скорость передачи данных, используемых для получения этих двух окон приема, идентичны тем, которые используются для окон приема RX1 и RX2.

Ответ оконечному устройству не передается, если запрос на подключение не принят.

Сообщение с присоединением (переприсоединением) к сети содержит поле случайного значения (JoinNonce) из 3 байтов, сетевой идентификатор (NetID), адрес оконечного устройства (DevAddr), поле с параметрами нисходящей линии связи (DLSettings), задержку между TX и RX (RxDelay) и дополнительный список сетевых параметров (CFList & CFListType) для сети, к которой присоединилось оконечное устройство (см. рисунок 55). Дополнительные поля CFList & CFListType являются региональными настройками и определены в разделе 9.

Размер (в байтах)	3	3	4	1	1	15	1
Подтверждение присоединения (переприсоединения) к сети	JoinNonce	Home_NetID	DevAddr	DLSettings	RxDelay	CFList	CFListType

Рисунок 55 — Структура сообщения с подтверждением присоединения (переприсоединения) к сети

JoinNonce содержит значение счетчика повторных присоединений для конкретного устройства (значения счетчика никогда не повторяются), предоставленное сервером присоединения (Join-Server) и используемое оконечным устройством для получения сеансовых ключей FNwkSIntKey, SNwkSIntKey, NwkSEncKey и AppSKey. JoinNonce увеличивается с каждым сообщением с подтверждением присоединения (переприсоединения) (Join-Accept).

Устройство отслеживает значение JoinNonce, использованное в последнем успешно обработанном Join-Accept (соответствует последней успешной генерации ключей). Устройство принимает Join-Accept, только если в поле MIC корректное значение и JoinNonce строго больше, чем записанное ранее. В этом случае новое значение JoinNonce заменяет ранее сохраненное.

Если устройство подвергается периодическому выключению/включению питания, JoinNonce при этом меняться не должен (должен сохраняться в энергонезависимой памяти).

Уникальный идентификатор сети (NetID) составляет 24 бита, за исключением следующих значений NetID, отведенных для экспериментальных/частных сетей, управление которыми не осуществляется.

Выделяется 2^{15} зарезервированных значений NetID для частных/экспериментальных сетей, формируемых согласно рисунку 56.

Биты	23:21	20:7	6:0
Значение	000	XXXXXXXXXXXXXX Произвольное 14-битное значение	0000000 или 0000001

Рисунок 56 — Формирование значений NetID для частных/экспериментальных сетей

Поле Home_NetID в Join-Accept соответствует NetID домашней сети устройства.

Сеть, которая присваивает DevAddr, и домашняя сеть могут быть разными в состоянии роуминга.

Поле «Параметры нисходящей линии связи» (DLsettings) содержит конфигурацию нисходящей линии связи согласно рисунку 57.

Биты	7	6:4	3:0
DLsettings	OptNeg	RX1DRoffset	RX2 Data rate

Рисунок 57 — Структура поля DLsettings

Бит OptNeg указывает, какую версию протокола реализует сетевой сервер: LoRaWAN v.1.0 (бит не установлен), v.1.1 и выше (бит установлен).

Когда бит OptNeg установлен, то:

- будет использоваться протокол версии LoRaWAN v.1.1 (или более поздний), соглашение между оконечным устройством и сетевым сервером осуществляется через обмен MAC-командами RekeyInd/RekeyConf;

- устройство вычисляет FNwkSIntKey & SNwkSIntKey & NwkSEncKey из NwkKey;
- устройство вычисляет AppSKey из AppKey.

Когда бит OptNeg не установлен, то:
- устройство использует LoRaWAN v.1.0;
- команда RekeyInd не отправляется устройством;
- устройство вычисляет FNwkSIntKey и AppSKey из NwkKey;
- устройство устанавливает значения ключей SNwkSIntKey и NwkSEncKey эквивалентными FNwkSIntKey.

Четыре сеансовых ключа FNwkSIntKey, SNwkSIntKey, NwkSEncKey и AppSKey формируются следующим образом.

Если OptNeg не установлен, то сеансовые ключи рассчитываются из NwkKey.

Сеансовый ключ приложения AppSKey рассчитывается с помощью первичного сеансового ключа NwkKey и сообщения:

$$message = 0x02 \,||\, JoinNonce \,||\, NetID \,||\, DevNonce \,||\, Pad_{16}$$

Функция Pad_{16} добавляет нулевой байт таким образом, чтобы длина данных стала кратной 16.

Непосредственный алгоритм кодирования не рассматривается в настоящем стандарте и выбирается в соответствии с требованиями других нормативных правовых актов, действующих на территории Российской Федерации.

Сетевой сеансовый ключ целостности восходящих сообщений FNwkSIntKey рассчитывается с помощью первичного сеансового ключа NwkKey и сообщения:

$$message = 0x01 \,||\, JoinNonce \,||\, NetID \,||\, DevNonce \,||\, Pad_{16}$$
$$SNwkSIntKey = NwkSEncKey = FNwkSIntKey$$

Значение MIC для сообщения подтверждения присоединения рассчитывается с помощью первичного сеансового ключа NwkKey и сообщения:

$$message = MHDR \,||\, JoinNonce \,||\, NetID \,||\, DevAddr \,||\, DLSettings \,||\, RxDelay \,||\, CFList \,||\, CFListType$$

Используются младшие 4 байта полученного значения.

Если OptNeg установлен, то AppSKey рассчитывается из AppKey следующим образом.

Сеансовый ключ приложения AppSKey рассчитывается с помощью прикладного первичного ключа AppKey и сообщения:

$$message = 0x02 \,||\, JoinNonce \,||\, JoinEUI \,||\, DevNonce \,||\, Pad_{16}$$

Сетевой сеансовый ключ FNwkSIntKey рассчитывается из NwkKey и сообщения:

$$message = 0x01 \,||\, JoinNonce \,||\, JoinEUI \,||\, DevNonce \,||\, Pad_{16}$$

Сетевой сеансовый ключ SNwkSIntKey рассчитывается из NwkKey и сообщения:

$$message = 0x03 \,||\, JoinNonce \,||\, JoinEUI \,||\, DevNonce \,||\, Pad_{16}$$

Сетевой сеансовый ключ NwkSEncKey рассчитывается из NwkKey и сообщения:

$$message = 0x04 \,||\, JoinNonce \,||\, JoinEUI \,||\, DevNonce \,||\, Pad_{16}$$

В этом случае значение MIC для сообщения подтверждения присоединения рассчитывается с помощью JSIntKey (ключ целостности восходящего сообщения с запросом Rejoin-Request первого типа и нисходящего сообщения ответа Join-Accept) и сообщения:

$$message = JoinEUI \,||\, DevNonce \,||\, MHDR \,||\, JoinNonce \,||\, NetID \,||\, DevAddr \,||\, DLSettings \,||\, RxDelay \,||$$
$$CFList \,||\, CFListType$$

Используются младшие 4 байта полученного значения.

JoinReqType представляет собой одно байтовое поле для кодирования типа запроса на присоединение (JoinRequest или RejoinRequest), инициировавшего ответ Join-Accept (таблица 16).

Таблица 16 — Кодирование JoinReqType

Тип JoinReqType	Значение JoinReqType
JoinRequest	0xFF
RejoinRequest type 0	0x00
RejoinRequest type 1	0x01
RejoinRequest type 2	0x02

Ключ, используемый для кодирования сообщения с подтверждением присоединения к сети, является функцией от сообщения с запросом присоединения (JoinRequest или RejoinRequest), инициировавшего его (таблица 17).

Таблица 17 — Значение ключа кодирования Join-Accept

Тип запроса	Ключ кодирования Join-Accept
JoinRequest	NwkKey
RejoinRequest type = 0 или 1 или 2	JSEncKey

Сообщение с подтверждением присоединения (переприсоединения) к сети (Join-Accept) кодируется с помощью ключа NwkKey (или JSEncKey) и сообщения:

message = JoinNonce || NetID || DevAddr || DLSettings || RxDelay || CFList || CFListType || MIC

Длина сообщения 16 или 32 байта.

Поле RX1DRoffset определяет смещение между скоростью передачи данных восходящего канала связи и скоростью передачи данных нисходящего канала связи, используемое для связи с оконечным устройством в первом окне приема (RX1). По умолчанию смещение равно 0. Смещение используется, чтобы учесть максимальные ограничения по мощности для базовых станций в некоторых регионах и для балансировки ограничений восходящей и нисходящей линий связи.

Фактические отношения между восходящей и нисходящей скоростью передачи данных определяются в региональных параметрах (приведены в разделе 9).

Задержка RxDelay следует тому же соглашению, что и поле Delay в команде *RXTimingSetupReq*.

Поля CFlist и CFlistType являются необязательными, но должны либо оба присутствовать, либо оба отсутствовать.

Если сообщение с подтверждением присоединения (переприсоединения) к сети (Join-Accept) получено после передачи:

- запроса на присоединение (или запроса на повторное присоединение типа 0 или 1) и если поле CFlist отсутствует, то устройство переходит на частотные каналы по умолчанию. Если CFlist присутствует, то он переопределяет все ранее заданные каналы. Параметры MAC-уровня (RXdelay1, скорость передачи данных RX2, ...) должны быть сброшены в значения по умолчанию;

- запроса на повторное присоединение типа 2 и если поле CFlist отсутствует, то устройство должно сохранить свои текущие частотные каналы без изменений. Если CFlist присутствует, то он переопределяет все определенные ранее каналы. Все остальные параметры MAC-уровня (за исключением счетчиков кадров, которые сбрасываются) остаются без изменения.

Во всех случаях после успешной обработки сообщения Join-Accept устройство передает MAC-команду *RekeyInd*, пока не получит команду *RekeyConf* (см. 6.3.10). Прием восходящей команды *RekeyInd* используется сетевым сервером как сигнал для перехода к новому сеансу.

6.4.2.4 Сообщение с запросом на повторное присоединение к сети (Rejoin-Request)

После активации устройство может периодически передавать запрос на повторное присоединение к сети (Rejoin-Request) (помимо обмена данными, определенного приложением). Это сообщение с запросом повторного присоединения дает возможность периодически, на стороне сервера, инициализировать новый сеанс для оконечного устройства. С этой целью сеть (сетевой сервер) отвечает сообщением подтверждения присоединения к сети (Join-Accept). Это может быть использовано для передачи (перемещения) устройства между двумя сетями или в исходной сети для замены ключей и/или изменения devAddr устройства.

Сетевой сервер может также использовать окна приема RX1/RX2 после запроса повторного присоединения для передачи нормальных подтвержденных или неподтвержденных нисходящих сообщений, дополнительно передавая MAC-команды. Эта возможность полезна для сброса параметров приема устройства в случае, если состояние MAC-уровня рассинхронизовалось между устройством и сетевым сервером.

Пример — Данный механизм может быть использован для изменения скорости передачи данных окна RX2 и смещения скорости передачи данных окна RX1 для устройства, которое более не доступно в нисходящей линии при использовании текущей конфигурации нисходящей линии связи.

Процедура повторного присоединения к сети всегда инициируется оконечным устройством путем отправки сообщения с запросом повторного присоединения.

Примечание — В любое время сетевой сервер обрабатывает запросы на повторное присоединение (типа 0,1 или 2) и генерирует сообщения с подтверждением присоединения к сети, он должен поддерживать как старый сеанс (ключи и счетчики, если они есть), так и новый, пока не получит первый успешный пакет из восходящей линии связи, используя новый сеанс, после чего старый сеанс следует удалить. Во всех случаях обработка сообщения с запросом на повторное присоединение сетевым сервером похожа на обработку стандартного сообщения с запросом на присоединение к сети, в котором сетевой сервер в начале обработки сообщения определяет, должен ли он передать его серверу присоединения (Join-Server) для формирования подтверждения присоединения к сети (Join-Accept) в ответ.

Существует три типа сообщений с запросом на повторное присоединение, которые могут быть переданы оконечным устройством и соответствуют трем различным целям. Первый байт сообщения с запросом на повторное присоединение называется «Тип повторного присоединения» (Rejoin Type) и используется для кодирования типа запроса на повторное присоединение. В таблице 18 описано назначение каждого типа сообщения с запросом на повторное присоединение к сети.

Таблица 18 — Типы сообщений с запросом на повторное присоединение к сети

Тип запроса	Описание и назначение
0	Содержит NetID+DevEUI. Используется для закрытия соединения на уровне сеанса с устройством, включая все параметры радио (devAddr, ключи сеанса, счетчики кадров, параметры радио). Такие сообщения могут направляться устройствами только на домашние сетевые сервера, не на сервер присоединения (JoinServer). MIC этого сообщения может быть проверен только при транзитном обслуживании или домашним сетевым сервером
1	Содержит JoinEUI+DevEUI. В точности эквивалентно исходному запросу присоединения Join-Request, но может передаваться поверх обычного прикладного трафика без отключения устройства. Только получающий его сетевой сервер может перенаправить на соответствующий устройству сервер Join. Используется для восстановления утерянного контекста сеанса (например, сетевой сервер потерял ключи сеанса и не может связать (ассоциировать) устройство с JoinServer). Только JoinServer способен проверить MIC этого сообщения
2	Содержит NetID+DevEUI. Используется для замены ключей устройства или изменения его devAddr (devAddr, сеансовые ключи, счетчики кадров). Параметры радио остаются неизменными. Эти сообщения могут направляться на домашний сетевой сервер только посещаемыми сетями (из роуминга). Не могут быть отправлены сервером присоединения Join оконечного устройства. MIC этого сообщения может быть проверен только при переправке (транзитном обслуживании) или домашним сетевым сервером

а) Сообщение с запросом на повторное присоединение типа 0 или 2

Сообщение с запросом на повторное присоединение типа 0 или 2 содержит NetID (идентификатор домашней сети устройства), DevEUI оконечного устройства и значение 16-битного счетчика (RJcount0) (см. рисунок 58).

Размер (в байтах)	1	3	8	2
ReJoin-Request	Rejoin Type = 0 или 2	NetID	DevEUI	RJcount0

Рисунок 58 — Структура сообщения с запросом на повторное присоединение

RJcount0 — счетчик, значение которого увеличивается с каждым переданным запросом на повторное присоединение типа 0 или 2. RJcount0 инициализируется в 0 каждый раз, когда подтверждение присоединения успешно обработано оконечным устройством. Для каждого оконечного устройства сетевой сервер должен отслеживать и хранить последнее значение RJcount0 (так называемый RJcount0_last), использованное оконечным устройством. Он игнорирует запросы на повторное присоединение, если (RJcount0 <= RJcount0_last).

RJcount0 никогда не повторяется (не должен использоваться в цикле при переполнении). Если RJcount0 достигает $2^{16} - 1$, устройство должно прекратить передачу запроса на повторное присоединение типа 0 или 2. Устройство может вернуться к состоянию присоединения к сети.

Примечание — Данный механизм предотвращает атаки посредством отправки предварительно записанных сообщений с запросами на повторное присоединение.

MIC для сообщения с запросом на повторное присоединение рассчитывается с помощью сетевого сеансового ключа целостности нисходящих сообщений SNwkSIntKey и сообщения:

Message = MHDR || Rejoin Type || NetID || DevEUI || RJcount0

Используются младшие 4 байта полученного значения.

Сообщение с запросом на повторное присоединение не кодируется.

Коэффициент «рабочий цикл устройства» (duty-cycle) при передаче запросов на повторное присоединение типа 0 или 2 должен быть менее 0,1 %.

Примечания

1 Сообщение с запросом на повторное присоединение типа 0 предполагается передавать (по замыслу) от одного раза в час до одного раза в несколько дней, в зависимости от варианта использования устройства. Это сообщение также может быть передано MAC-командой *ForceRejoinReq*. Это сообщение может использоваться для переподключения мобильного устройства к гостевой сети при роуминге. Также может быть использовано для замены ключей или изменения devAddr статического устройства. Мобильные устройства, как ожидается, перемещаясь между сетями, должны передавать это сообщение чаще, чем статические устройства.

2 Сообщение с запросом на повторное присоединение типа 2 предназначено только для обеспечения замены ключей оконечного устройства. Это сообщение может передаваться только MAC-командой *ForceRejoinReq*.

б) Сообщение с запросом на повторное присоединение типа 1

Аналогично запросу на присоединение сообщение с запросом на повторное присоединение типа 1 содержит JoinEUI и DevEUI оконечного устройства (см. рисунок 59). Поэтому сообщение с запросом на повторное присоединение типа 1 может быть направлено серверу присоединения оконечного устройства (Join-Server) любым сетевым сервером, принявшим его. Запрос на повторное присоединение типа 1 может использоваться для восстановления связи с оконечным устройством в случае полной потери сетевого сервера. Рекомендуется передавать сообщение с запросом на повторное присоединение типа 1 не реже одного раза в месяц.

Размер (в байтах)	1	8	8	2
ReJoin-Request	Rejoin Type = 1	JoinEUI	DevEUI	RJcount1

Рисунок 59 — Структура сообщения с запросом на повторное присоединение к сети типа 1

RJcount1 для запроса на повторное присоединение к сети типа 1 — это другой счетчик, не RJCount0 (который используется для запроса на повторное присоединение типа 0).

RJcount1 — счетчик, значение которого увеличивается с каждым переданным запросом на повторное присоединение типа 1. Для каждого оконечного устройства сервер присоединения (Join-Server) отслеживает и хранит последнее значение RJcount1 (так называемый RJcount1_last), использованное оконечным устройством. Он игнорирует запросы на повторное присоединение, если (RJcount1 <= RJcount1_last).

RJcount1 никогда не повторяется для выданного JoinEUI. Периодичность отправки запроса на повторное присоединение типа 1 должна быть такой, чтобы не могло произойти переполнение счетчика и повторное использование его значений в период жизни устройства с заданным значением JoinEUI.

Примечание — Данный механизм предотвращает атаки посредством отправки предварительно записанных сообщений с запросами на повторное присоединение.

MIC для сообщения с запросом на повторное присоединение типа 1 рассчитывается с помощью ключа JSIntKey (ключ кодирования Join-Accept в случае получения запроса Rejoin-Request) и сообщения:

Message = MHDR || Rejoin Type || JoinEUI || DevEUI || RJcount1

Используются младшие 4 байта полученного значения.

Сообщение с запросом на повторное присоединение типа 1 не кодируется.

Рабочий цикл устройства (duty-cycle) при передаче запросов на повторное присоединение типа 1 всегда должен быть <0.01%.

Примечание — Сообщение с запросом на повторное присоединение типа 1 предполагается передавать от одного раза в день до одного раза в неделю. Это сообщение используется только в случае полной потери сервером контекста уровня сеанса. Это событие очень маловероятно, поэтому повторное подключение устройства с периодичностью от одного раза в день до одного раза в неделю считается приемлемым.

в) Передача запроса на повторное присоединение

В таблице 19 приведены возможные условия для передачи сообщения каждого типа запроса на повторное присоединение.

Таблица 19 — Возможные условия для передачи сообщения каждого типа запроса на повторное присоединение

Тип запроса на повторное присоединение (RejoinRequest)	Передается автономно и периодически оконечным устройством	Передается с MAC-командой ForceRejoinReq
0	x	x
1	x	
2		x

Сообщение с запросом на повторное присоединение типов 0 и 1 должно передаваться по любому определенному для процедуры присоединения каналу (см. раздел 9) на соответствующих случайно переключаемых (выбираемых) частотах.

Запрос на повторное присоединение типа 2 должен передаваться по любому включенному в настоящий момент каналу на соответствующих случайно переключаемых (выбираемых) частотах.

Запросы на повторное присоединение типов 0 и 2, передаваемые с помощью команды ForceRejoinReq, должны использовать скорость передачи данных, указанную в MAC-команде.

Запросы на повторное присоединение типа 0 [передаются периодически и автономно оконечным устройством (с максимальной периодичностью, установленной командой RejoinParamSetupReq)] и запросы на повторное присоединение типа 1 должны использовать:

- скорость передачи данных и выходную мощность передатчика, используемые в настоящий момент для передачи прикладных данных (payload), если включен ADR;
- любую разрешенную для каналов присоединения скорость передачи данных и мощность передатчика по умолчанию, если ADR отключена. В этом случае рекомендуется использовать различные скорости передачи данных.

г) Обработка запроса на повторное присоединение

Для всех трех типов запроса на повторное присоединение сетевой сервер может реагировать:

- сообщением с подтверждением присоединения (как описано в 6.4.2.3), если он хочет изменить сетевой идентификатор устройства (роуминг или замена ключей). В этом случае RJcount (0 или 1) заменяет DevNonce в процедуре получения ключей;
- нормальным (обычным) нисходящим кадром (сообщением), дополнительно содержащим MAC-команды. Этот нисходящий кадр должен быть отправлен в тот же канал с той же скоростью передачи данных и с той же задержкой, что были заданы для сообщения с подтверждением подключения, которое он заменяет.

В большинстве случаев на попутные запросы на повторное присоединение типа 0 или 1 сеть не будет реагировать.

6.5 Задержка повторных передач

Для восходящих кадров, для которых одновременно выполняются условия (1) и (3) или (2) и (3), существует ограничение на загрузку радиоэфира восходящими сообщениями. Условия, при которых действуют ограничения:

1) требующих подтверждения или ответа от сервера сети или сервера приложений;

2) являющихся повторной передачей по причине отсутствия ответа (подтверждения) от сервера;

3) объединенных внешним событием (отключение электричества, отключение сети и т. д.), которое может инициировать одновременную синхронизацию большого количества устройств (>100), что может вызвать катастрофическую, тяжело восстанавливаемую ситуацию перегрузки радиосети.

Примечание — Примером такого восходящего кадра является JoinRequest, когда он выполняется группой оконечных устройств, решивших осуществить сброс MAC-уровня в случае сбоя сети. Вся группа оконечных устройств начинает отправлять восходящие JoinRequest и прекращает только после получения от сети JoinResponse.

Для таких повторных отправок кадров интервал между окончанием окна приема RX2 и следующей повторной передачей в восходящую линию связи должен быть случайным для каждого устройства (например, рекомендуется использование генератора псевдослучайных чисел с адресом устройства). Рекомендуется, чтобы рабочий цикл передачи таких сообщений соответствовал региональным параметрам (см. раздел 9) и приведенным в таблице 20 ограничениям, в зависимости от того, чьи ограничения более строгие.

Таблица 20 — Требования к допустимой загрузке радиоэфира

Описание периода	Момент передачи очередного кадра (t)	Допустимая загрузка радиоэфира
В течение первого часа после включения питания или сброса	$T_0 < t < T_0 + 1h$	Время передачи < 36 секунд за 1 час
После первого часа, в течение следующих 10 часов	$T_0 + 1h < t < T_0 + 11h$	Время передачи < 36 секунд за 11 часов
После первых 11 часов, в течение следующих 24 часов и за каждые последующие 24 часа	$T_0 + 11 + N < t < T_0 + 35 + N$ $N \geq 0$	Время передачи < 8,7 секунд за 24 часа

7 Оконечные устройства класса C

7.1 Режим связи оконечного устройства

Оконечные устройства класса C используются там, где есть возможность использовать внешний источник питания (питаются от сети постоянного питания) и, следовательно, не требуется минимизировать время приема.

Оконечное устройство класса C большую часть времени прослушивает радиоэфир с параметрами окна приема RX2. Оконечное устройство должно слушать в окне приема RX2, когда оно не передает (а) либо не принимает в окне приема RX1 (б) в соответствии с описанием класса A. Для этого ему необходимо открыть маленькое (короткое) окно, использующее параметры RX2, между концом передачи в восходящую линию связи и началом окна приема RX1 и необходимо переключиться на параметры окна приема RX2; как только окно приема RX1 закроется, окно приема RX2 должно оставаться открытым до тех пор, пока оконечному устройству не потребуется послать еще одно сообщение.

Примечания

1 Если устройство находится в процессе демодуляции нисходящего сообщения, используя параметры RX2, в момент, когда должно быть открыто окно приема RX1, то устройство должно прекратить демодуляцию и переключиться на прием в окне RX1.

2 Устройство класса C не может сообщить серверу, что оно поддерживает класс C. Сведения о принадлежности устройства к классу C должны попадать в сервер с прикладного уровня.

В случае, если сообщение принимается устройством, работающим в режиме класса C, и требуется передача восходящего сообщения (нисходящая MAC-команда — запрос или нисходящее сообщение, требующее подтверждения), устройство должно ответить в течение периода времени, известного как оконечному устройству, так и сетевому серверу.

До истечения этого периода (тайм-аута) сеть не должна направлять какие-либо новые сообщения, требующие подтверждения, или MAC-команды на устройство. После истечения этого периода или после приема любого восходящего сообщения сети разрешено посылать новое нисходящее сообщение.

7.1.1 Длительность второго окна приема для класса С

Устройства класса С реализуют те же два окна приема, что и устройства класса А, но они не закрывают окно приема RX2 до момента отправки очередного восходящего сообщения (см. рисунок 60). Поэтому они могут получать нисходящие сообщения в окне приема RX2 почти в любое время, в том числе нисходящие сообщения, отправленные с целью передачи MAC-команды или подтверждения получения сообщения (ACK). Короткое окно прослушивания на частоте и скорости передачи данных RX2 также открывается между окончанием передачи и началом приема в окне RX1.

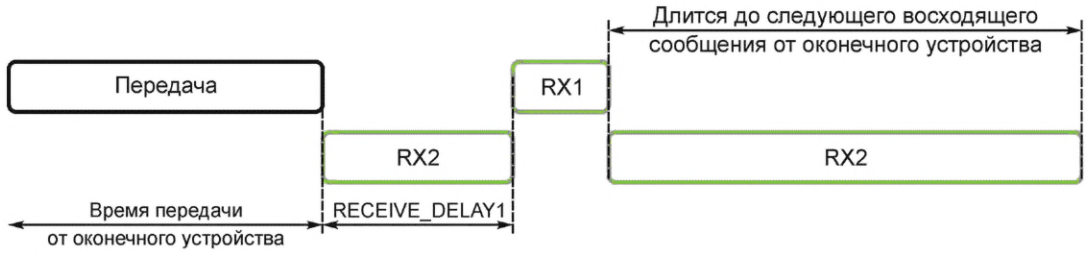

Рисунок 60 — Временной график приема сообщений для класса С

7.1.2 Многоадресная рассылка для класса С

Устройства класса С могут принимать многоадресные нисходящие пакеты. Адрес многоадресной рассылки и соответствующие сетевой сеансовый ключ и сеансовый ключ приложения должны приходить на уровне приложения.

Примечание — Многоадресная рассылка может использоваться для многоадресной передачи следующих данных: обновление встроенного программного обеспечения, единое время, альманах и эфемериды GPS/GLONASS-спутников (для ускоренного определения координат оконечными устройствами) и т. д.

Ограничения, распространяющиеся на многоадресные нисходящие сообщения для класса С:

- сообщения передаются только в нисходящем канале связи;
- сообщения не должны нести MAC-команды ни в области FOpts, ни в поле прикладных данных FRMPayload на порт 0;
- биты ACK и ADRACKReq должны быть равны 0;
- поле MType должно нести значение, соответствующее нисходящему сообщению, не требующему подтверждения (MType = Unconfirmed Data Down);
- бит FPending должен указывать на то, что имеются еще многоадресные данные для отправки.

Примечание — Учитывая, что устройство класса С сохраняет активным свой приемник большую часть времени, бит FPending не вызывает какого-либо конкретного поведения оконечного устройства.

7.2 MAC-команды

Все команды, описанные для класса А, должны быть реализованы в устройствах класса С. Для устройств класса С дополнительно определены MAC-команды, указанные в таблице 21.

Таблица 21 — MAC-команды для устройств класса С

CID	Команда	Передается		Краткое описание
		КУ	БС	
0x20	DeviceModeInd	x		Используется оконечным устройством для обозначения его текущего режима работы (класс А или С)
0x20	DeviceModeConf		x	Используется сетью для подтверждения команды DeviceModeInd

7.2.1 Режим работы устройства (*DeviceModeInd, DeviceModeConf*)

С помощью команды *DeviceModeInd* оконечное устройство извещает сеть о режиме своей работы в классе А или С. Команда имеет данные размером один байт (см. рисунок 61).

Размер (в байтах)	1
DeviceModeInd Payload	Класс (Class)

Рисунок 61 — Атрибут команды *DeviceModeInd*

Значения классов для команды *DeviceModeInd* приведены в таблице 22.

Таблица 22 — Значения классов для команды *DeviceModeInd*

Поле «Класс» (Class)	Значение
Класс А (Class A)	0x00
RFU	0x01
Класс С (Class C)	0x02

Когда сетевой сервер получает команду *DeviceModeInd*, он отвечает на нее командой *DeviceModeConf*. Устройство должно включать команду *DeviceModeInd* во все восходящие сообщения, пока не получит команду *DeviceModeConf*.

Устройство должно переключить режим работы, как только первая команда *DeviceModeInd* будет передана.

Примечание — Для устройств с батарейным питанием рекомендуется при переходе от класса А к классу С реализовать механизм тайм-аутов на прикладном уровне, чтобы гарантировать, что устройство не задержится на неопределенный срок в режиме класса С при отсутствии связи с сетью.

Команда *DeviceModeConf* содержит один байт данных (рисунок 62).

Размер (в байтах)	1
Данные DeviceModeConf (DeviceModeConf Payload)	Class

Рисунок 62 — Структура команды *DeviceModeConf*

Параметр Class определяется так же, как для MAC-команды *DeviceModeInd*.

8 Примеры реализации

Ниже приведены примеры, иллюстрирующие применение настоящего стандарта.

8.1 Диаграмма передачи восходящего сообщения с подтверждением

Следующая диаграмма (см. рисунок 63) иллюстрирует шаги, выполняемые оконечным устройством, которое пытается передать два восходящих сообщения (Data0 и Data1) с требованием подтверждения. Параметр NbTrans этого устройства должен быть больше или равен 2, чтобы этот пример был действительным (т. к. первый подтвержденный кадр передается дважды).

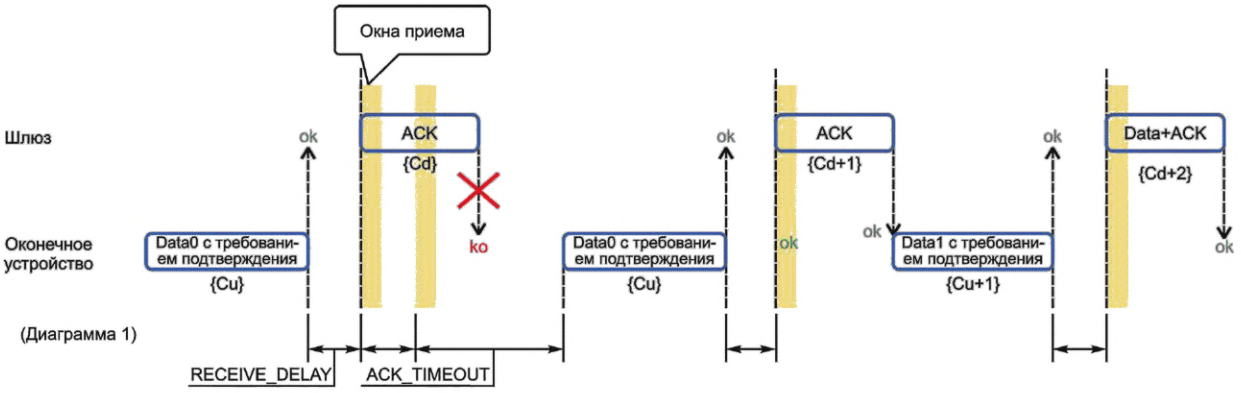

Рисунок 63 — Диаграмма передачи восходящего сообщения с подтверждением

Оконечное устройство сначала передает кадр данных, требующий подтверждения, содержащий данные Data0, в произвольный момент времени и на произвольном канале. Значение счетчика кадров (Cu) формируется путем добавления 1 к предыдущему счетчику кадров восходящей линии связи. Сервер сети принимает кадр и генерирует кадр в нисходящую линию связи. В нисходящем сообщении установлен бит ACK, т. е. подтверждение получения предыдущего сообщения. Нисходящее сообщение передается с задержкой RECEIVE_DELAY в первое окно приема RX1 оконечного устройства. Данное нисходящее сообщение использует ту же скорость передачи данных и тот же частотный канал, что и предыдущее восходящее сообщение с Data0. Счетчик кадров в нисходящей линии связи (Cd) получается путем добавления 1 к предыдущему значению счетчика кадров в нисходящей линии для данного экземпляра оконечного устройства. Если в сервере нет данных, ожидающих передачи в оконечное устройство, то сеть должна генерировать сообщение без прикладных данных. В данном примере кадр, несущий бит ACK, не принимается оконечным устройством из-за помех в радиоканале.

Если оконечное устройство не получает в течение времени ACK_TIMEOUT ни в одном из окон приема (RX1 или RX2) кадр с битом ACK, то оконечное устройство может повторно отправить те же данные (Data0) с тем же счетчиком кадров (Cu). Эта повторная отправка должна выполняться на другом частотном канале и должна соответствовать ограничению рабочего цикла (DutyCycle), как и любая другая передача в восходящем канале. Если на этот раз оконечное устройство принимает в нисходящем канале подтверждение (бит ACK) во время своего первого окна приема RX1, то оконечное устройство затем может передавать следующий кадр (Data1) на новый канал.

Кадр нисходящей линии связи может нести комбинацию сведений: подтверждение предыдущего сообщения (ACK), MAC-команды и прикладные данные.

8.2 Диаграмма передачи нисходящего сообщения с подтверждением

Следующая диаграмма (см. рисунок 64) иллюстрирует типовую последовательность передачи сообщения с подтверждением в нисходящую линию связи.

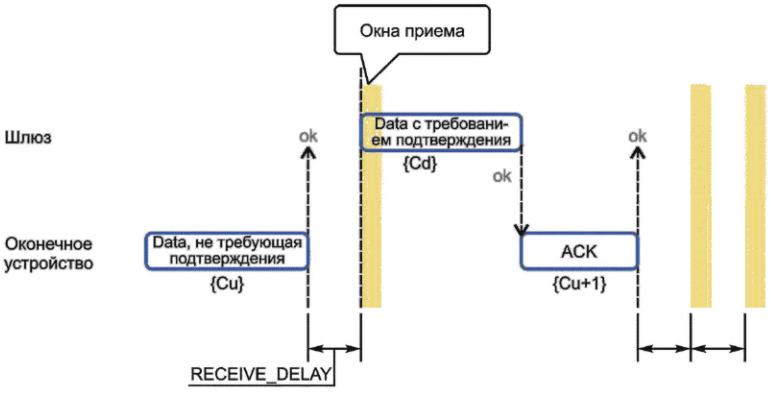

Рисунок 64 — Диаграмма передачи нисходящего сообщения с подтверждением

Обмен данными инициируется оконечным устройством класса А, передающим сообщение (Data), не требующее подтверждения. Сервер сети использует первое окно приема (RX1) в нисходящей линии связи для передачи сообщения, требующего подтверждения в направлении оконечного устройства. Передача нисходящего сообщения осуществляется на канале предыдущего восходящего сообщения. Оконечное устройство после приема данного нисходящего сообщения передает сообщение с битом ACK, подтверждающим получение предыдущего сообщения. Данное восходящее сообщение может содержать данные или MAC-команды и передается на новом канале, выбранном случайным образом.

Примечание — Чтобы оконечные устройства были максимально простыми и имели как можно меньше состояний, они могут передавать пустое сообщение с подтверждением (без прикладных данных) сразу после приема нисходящего сообщения, требующего подтверждения. В качестве альтернативы оконечное устройство может отложить передачу подтверждения, чтобы передать его вместе со своими следующими прикладными данными.

8.3 Диаграмма передачи очереди нисходящих сообщений

Следующая диаграмма иллюстрирует управление очередью нисходящих сообщений с помощью бита FPending.

Бит FPending может быть установлен только в кадре нисходящей линии связи и информирует оконечное устройство о том, что сервер имеет несколько сообщений в очереди для передачи данному оконечному устройству. В восходящих сообщениях бит FPending игнорируется сервером сети.

Если кадр с установленным битом FPending=1 требует подтверждения, то оконечное устройство должно сделать это, как описано выше (8.2). Если подтверждение не требуется, оконечное устройство может отправить пустое сообщение (без прикладных данных), чтобы открыть очередные окна приема (RX1 и RX2) или дождаться, когда в оконечном устройстве появятся прикладные данные, которые необходимо передать в сервер сети.

Примечание — Бит FPending не зависит от подтверждения (ACK).

Пример 1 приведен на рисунке 65.

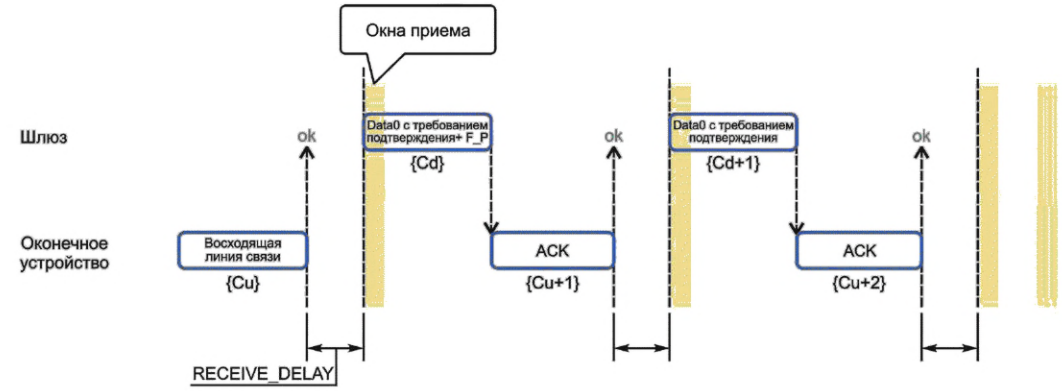

Рисунок 65 — Диаграмма передачи очереди нисходящих сообщений (пример 1)

В данном примере сеть передает в оконечное устройство два сообщения, требующих подтверждения.

Обмен сообщениями инициируется оконечным устройством класса А посредством передачи сообщения в восходящую линию связи. Сообщение передается на частотном канале chA. Сеть использует первое окно приема (RX1) для передачи на канале chA данных Data0 с установленными битом FPending и требованием подтверждения. Устройство, получив сообщение с битом FPending=1, передает на новом частотном канале chB подтверждение приема данного сообщения, передавая обратно пустой кадр с битом ACK.

С задержкой в RECEIVE_DELAY1 секунд сеть на канале chB передает в устройство второе сообщение (Data1), требуя подтвердить получение сообщения, но с битом FPending, теперь равным 0.

Оконечное устройство подтверждает получение (Data1) на канале chC.

Пример 2 приведен на рисунке 66.

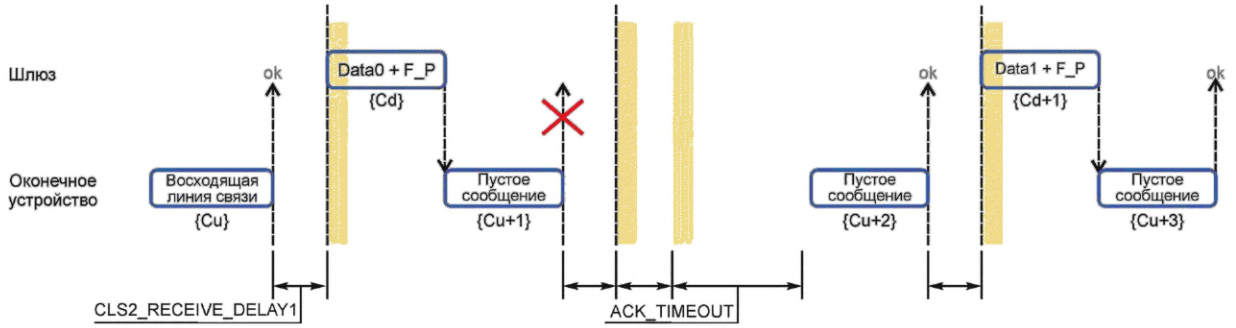

Рисунок 66 — Диаграмма передачи очереди нисходящих сообщений (пример 2)

В данном примере сообщения в нисходящей линии связи являются сообщениями, не требующими подтверждения оконечным устройством.

Оконечное устройство при получении сообщения Data0, не требующего подтверждения, но с установленным битом FPending, отправляет в сеть пустое сообщение без прикладных данных. Это первое восходящее сообщение не принимается сетью по причине помех. Если ни одно нисходящее сообщение не было получено в течение двух последующих окон приема (RX1 и RX2), то оконечное устройство должно повторить передачу пустого восходящего сообщения. После получения сервером сети пустого сообщения в одно из окон приема (RX1 и RX2) отправляется следующее нисходящее сообщение (cd+1) из очереди.

Примечание — Подтверждение никогда не отправляется дважды.

Пример 3 приведен на рисунке 67.

Бит FPending, бит ACK и прикладные данные могут одновременно присутствовать в одном нисходящем сообщении.

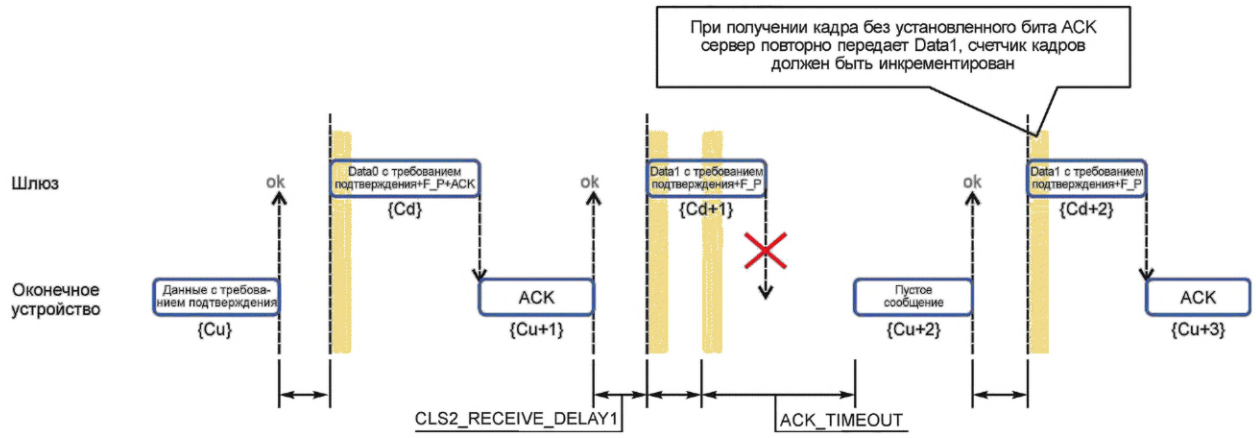

Рисунок 67 — Диаграмма передачи очереди нисходящих сообщений (пример 3)

Оконечное устройство отправляет в восходящую линию связи данные, требующие подтверждения. Сервер сети может ответить сообщением, содержащим: подтверждение получения сообщения из восходящей линии связи, данные нисходящей линии связи Data, требующие подтверждения, и поле FPending=1, информирующее о наличии очереди сообщений для данного устройства.

9 Региональные параметры

9.1 RU864-870

9.1.1 Формат поля «Преамбула»

Структура поля «Преамбула» приведена в таблице 23.

Таблица 23 — Структура поля «Преамбула»

Модуляция	Синхрослово	Размер поля «Преамбула»
ЛЧМ	0x34	8 символов
GFSK	0xC194C1	5 байт

9.1.2 Частотные каналы

Используемые каналы связи должны соответствовать требованиям регламентирующих документов Государственной комиссии по радиочастотам.

Все доступные частотные каналы могут использоваться оператором связи по его усмотрению. Два канала «по умолчанию» должны быть реализованы в каждом оконечном устройстве (см. таблицу 24). Данные каналы являются обязательными и не могут быть отредактированы MAC-командой NewChannelReq. Эти каналы являются минимальным набором, который должен всегда прослушиваться всеми радиошлюзами сети связи.

Таблица 24 — Каналы «по умолчанию»

Номер канала	Модуляция	Частота канала, МГц	Полоса частот, кГц	Скорость	Рабочий цикл (DutyCycle)	Мощность, мВт/дБм
1	ЛЧМ	868,9	125	DR0 … DR5	<10 %	25/14
2	ЛЧМ	869,1	125	DR0 … DR5	<10 %	25/14

Каналы, используемые оператором связи по его усмотрению, приведены в таблице 25.

Таблица 25 — Каналы, используемые оператором связи по его усмотрению

Номер канала	Модуляция	Частота канала, МГц	Полоса частот, кГц	Скорость	Рабочий цикл (DutyCycle)	Мощность, мВт/дБм
3	ЛЧМ	864,1	125	DR0 … DR5	0,1 % или LBT	25/14
4	ЛЧМ	864,3	125	DR0 … DR5	0,1 % или LBT	25/14
5	ЛЧМ	864,5	125	DR0 … DR5	0,1 % или LBT	25/14
6	ЛЧМ	864,7	125	DR0 … DR5	0,1 % или LBT	25/14
7	ЛЧМ	864,9	125	DR0 … DR5	0,1 % или LBT	25/14
8	ЛЧМ	866,1	125	DR0 … DR5	1 % или LBT	25/14
9	ЛЧМ	866,3	125	DR0 … DR5	1 % или LBT	25/14
10	ЛЧМ	866,5	125	DR0 … DR5	1 % или LBT	25/14
11	ЛЧМ	866,7	125	DR0 … DR5	1 % или LBT	25/14
12	ЛЧМ	866,9	125	DR0 … DR5	1 % или LBT	25/14
13	ЛЧМ	867,1	125	DR0 … DR5	1 % или LBT	25/14
14	ЛЧМ	867,3	125	DR0 … DR5	1 % или LBT	25/14
15	ЛЧМ	867,5	125	DR0 … DR5	1 % или LBT	25/14
16	ЛЧМ	867,7	125	DR0 … DR5	1 % или LBT	25/14
17	ЛЧМ	867,9	125	DR0 … DR5	1 % или LBT	25/14

Примечание — LBT — функция прослушивания радиоэфира перед передачей.

Оконечные устройства должны обеспечивать возможность хранения не менее семи частотных каналов.

Примечание — Под каналом понимаются частота и диапазон скоростей, доступных для передачи на данной частоте.

Список каналов, доступных для передачи запроса на присоединение к сети, приведен в таблице 26.

Таблица 26 — Список каналов, доступных для передачи запроса на присоединение к сети

Номер канала	Модуляция	Частота канала, МГц	Полоса частот, кГц	Скорость	Рабочий цикл (DutyCycle)	Мощность, мВт/дБм
1	ЛЧМ	868,9	125	DR0 … DR5	<10 %	25/14
2	ЛЧМ	869,1	125	DR0 … DR5	<10 %	25/14

9.1.3 Кодирование скорости и мощности

Кодирование скорости передачи данных (DataRate) приведено в таблице 27.

Таблица 27 — Кодирование скорости передачи данных

Скорость передачи данных (DataRate)	Конфигурация	Физическая скорость передачи данных, бит/с
0	ЛЧМ: SF12/125 кГц	250
1	ЛЧМ: SF11/125 кГц	440
2	ЛЧМ: SF10/125 кГц	980
3	ЛЧМ: SF9/125 кГц	1760
4	ЛЧМ: SF8/125 кГц	3125
5	ЛЧМ: SF7/125 кГц	5470
6	ЛЧМ: SF7/250 кГц	11000
7	FSK: 50 кбит/с	50000
8..14	Зарезервировано	Зарезервировано
15	Сохранить предыдущее значение	

Кодирование выходной мощности оконечного устройства (TXPower) приведено в таблице 28.

Таблица 28 — Кодирование выходной мощности оконечного устройства

TXPower	Физическое значение
0	27 дБм (Зарезервировано)
1	20 дБм (Зарезервировано)
2	16 дБм (Зарезервировано)
3	14 дБм
4	12 дБм
5	10 дБм
6	8 дБм
7	6 дБм
8	4 дБм
9	2 дБм
От 10 до 14	Зарезервировано
15	Сохранить предыдущее значение

Значения «Сохранить предыдущее значение» используются в MAC-команде LinkADRReq при редактировании диапазона допустимых скоростей и мощности передачи пакета.

9.1.4 Частотные каналы, передаваемые в CFList в подтверждение присоединения (Join-Accept)

В поле CFlist (см. рисунок 68) передается в сообщении Join-Accept и содержит дополнительный список из пяти частотных каналов длиною 15 байт. За списком частот следует 1 байт CFListType, всего 16 байт. CFListType должен быть равен нулю (0), чтобы указать, что CFList содержит список частот.

Каждая частота кодируется как 24-разрядное целое число без знака (3 байта). Все каналы могут использоваться со скоростью от DR0 до DR5 в полосе 125 кГц.

Размер (байт)	3	3	3	3	3
CFList	Freq ch3	Freq ch4	Freq ch5	Freq ch6	Freq ch7

Рисунок 68 — Структура поля CFList

Фактическая частота канала в Гц равна 100.[Freq chX]. Это позволяет установить частоту канала в диапазоне от 100 МГц до 1,67 ГГц с шагом 100 Гц.

Неиспользуемые каналы имеют значение частоты, равное 0.

Поле CFList является необязательным в сообщении «Подтверждение присоединения», и его присутствие может быть выявлено по длине сообщения «Подтверждение присоединения». Если поле CFList присутствует, то устройство заменяет все предыдущие каналы, хранящиеся в оконечном устройстве, кроме двух каналов «по умолчанию». Новые каналы сразу могут быть использованы оконечным устройством.

9.1.5 Маска каналов в MAC-команде *LinkAdrReq*

Когда значение поля «Управление маской канала» (ChMaskCntl) равно 0, то поле «Маска канала» (ChMask) индивидуально включает/выключает каждый от 1 до 16 каналов (см. таблицу 29).

Таблица 29 — Маска каналов в MAC-команде

Значение поля «Управление маской канала» (ChMaskCntl)	Область применения поля «Маска канала» (ChMask)
0	Каналы от 1 до 16
1	Зарезервировано
…	...
4	Зарезервировано
5	Зарезервировано
6	Все каналы включены. Устройство должно включить все известные каналы, независимо от значения поля «Маска канала» (ChMask)
7	Зарезервировано

Если значение поля «Управление маской канала» (ChMaskCntl) «Зарезервировано», то оконечное устройство должно отклонить MAC-команду и отключить бит «ACK» в ответе.

9.1.6 Максимальный размер поля данных (MACPayload)

Максимальный размер поля данных (M) приведен в таблице 30. Он получен из ограничений физического уровня в зависимости от эффективной скорости модуляции. Максимальная длина прикладных данных (FRMPayload) в отсутствие дополнительного поля управления FOpt (N) предоставляется только для информации. Значение N может быть соответственно меньше, если поле FOpt содержит данные длиною от 1 до 15 байт.

Таблица 30 — Максимальный размер поля данных

DataRate	M	N
0	59	51
1	59	51
2	59	51
3	123	115
4	230	222
5	230	222
6	230	222
7	230	222
От 8 до 15	Не определено	Не определено

9.1.7 Окна приема RX1/RX2

В окне приема RX1 должен использоваться тот же частотный канал, который использовался при передаче предыдущего восходящего сообщения.

Скорость передачи данных в окне RX1 зависит от скорости передачи данных по восходящей линии связи и значения RX1DROffset согласно таблице 31.

Таблица 31 — Скорость передачи данных в окне RX1

Скорость в восходящем канале	Скорость в приемном окне RX1					
	В зависимости от значения RX1DROffset					
	0	1	2	3	4	5
DR0	DR0	DR0	DR0	DR0	DR0	DR0
DR1	DR1	DR0	DR0	DR0	DR0	DR0
DR2	DR2	DR1	DR0	DR0	DR0	DR0
DR3	DR3	DR2	DR1	DR0	DR0	DR0
DR4	DR4	DR3	DR2	DR1	DR0	DR0
DR5	DR5	DR4	DR3	DR2	DR1	DR0

Допустимые значения для RX1DROffset находятся в диапазоне от 0 до 5. Значения в диапазоне от 6 до 7 зарезервированы для будущего использования.

В приемном окне RX2 используются фиксированная частота и скорость передачи данных. Параметры по умолчанию: 869,1 МГц на скорости DR0 (SF12, 125 кГц).

9.1.8 Настройки по умолчанию

Параметры, указанные в таблице 32, рекомендуется использовать по умолчанию.

Таблица 32 — Настройки по умолчанию

Параметр	Значение
RECEIVE_DELAY1	1 с
RECEIVE_DELAY2	RECEIVE_DELAY1 + 1
JOIN_ACCEPT_DELAY1	5 с
JOIN_ACCEPT_DELAY2	6 с
MAX_FCNT_GAP	16384
ADR_ACK_LIMIT	64
ADR_ACK_DELAY	32
ACK_TIMEOUT	1…3 с (случайное значение в интервале)

Если фактические значения параметров, реализованные в оконечном устройстве, отличаются от значений по умолчанию (например, оконечное устройство использует более короткую задержку RECEIVE_DELAY1 и RECEIVE_DELAY2), то эти параметры должны быть переданы серверу сети во время ввода в эксплуатацию оконечного устройства. Сервер сети может не принимать значения параметров, отличных от указанных по умолчанию.

Библиография

[1] IEEE Standard for Local and Metropolitan Area Networks — Part 15.4: Low-2699 Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4TM-2011 (Revision 2700 of IEEE Std 802.15.4-2006), September 2011.

УДК 004.738:006.354

ОКС 35.020
35.110

Ключевые слова: информационные технологии, интернет вещей, LoRaWAN RU

Редактор *Л.В. Коретникова*
Технический редактор *И.Е. Черепкова*
Корректор *Л.С. Лысенко*
Компьютерная верстка *И.Ю. Литовкиной*

Сдано в набор 25.12.2023. Подписано в печать 15.01.2024. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 6,51. Уч-изд. л. 5,70.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru