ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПНСТ 354— 2019

Информационные технологии

ИНТЕРНЕТ ВЕЩЕЙ

Протокол беспроводной передачи данных на основе узкополосной модуляции радиосигнала (NB-Fi)

Издание официальное

Москва
Стандартинформ
2019

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Обществом с ограниченной ответственностью «Телематические Решения» (ООО «Телематические Решения»), Акционерным обществом «Российская венчурная компания» (АО «РВК»), Фондом развития интернет-инициатив (ФРИИ), Ассоциацией участников рынка интернета вещей и Некоммерческим партнерством «Русское общество содействия развитию биометрических технологий, систем и коммуникаций» (Некоммерческое партнерство «Русское биометрическое общество»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 194 «Киберфизические системы»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 19 февраля 2019 г. № 7-пнст

Правила применения настоящего стандарта и проведения его мониторинга установлены в ГОСТ Р 1.16—2011 (разделы 5 и 6).

Федеральное агентство по техническому регулированию и метрологии собирает сведения о практическом применении настоящего стандарта. Данные сведения, а также замечания и предложения по содержанию стандарта можно направить не позднее чем за 4 мес до истечения срока его действия разработчику настоящего стандарта по адресу: 143026 Москва, территория инновационного центра Сколково, ул. Нобеля, д. 5, пом. 334, тел. 8 (800) 550-51-89, e-mail: gost.e-ecolog.ru *и в Федеральное агентство по техническому регулированию и метрологии: 109074 Москва, Китайгородский проезд, д. 7, стр. 1.*

*В случае отмены настоящего стандарта соответствующая информация будет опубликована в ежемесячном информационном указателе «Национальные стандарты» и также будет размещена на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, оформление, 2019

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 . . . 1
3 Термины и определения . . . 1
4 Сокращения . . . 2
5 Физический уровень (Physical layer) . . . 2
5.1 Общие положения . . . 2
5.2 UPLINK-пакет . . . 2
5.3 DOWNLINK-пакет . . . 3
5.4 Режим работы LBT (Listen Before Talk) . . . 4
6 MAC-уровень (Media Access Control layer, MAC) . . . 4
6.1 Общие положения . . . 4
6.2 UPLINK-пакет . . . 5
6.3 DOWNLINK-пакет . . . 6
6.4 Шифрование поля Payload . . . 6
7 Транспортный уровень (Transport layer) . . . 6
7.1 Общие положения . . . 6
7.2 Описание пакетов данных транспортного уровня . . . 7
8 Уровень представления (Presentation layer) . . . 23
8.1 Общие положения . . . 23
8.2 Описание пакетов данных уровня представления . . . 24
Приложение А (справочное) Описание алгоритма компенсации нестабильности частот задающего генератора . . . 26
Приложение Б (справочное) Фрагменты исходных кодов реализации MAC-уровня . . . 28
Приложение В (справочное) Таблица и функции, используемые для ZIGZAG-кодирования данных . . . 31
Приложение Г (справочное) Описание транспортных функций и сценариев взаимодействия . . . 33
Приложение Д (справочное) Описание механизма автоматического выбора оптимальной скорости передачи данных . . . 37
Приложение Е (справочное) Настройка параметров NB-Fi . . . 39
Приложение Ж (справочное) Защита данных в протоколе NB-Fi . . . 40
Приложение И (справочное) Примеры использования протокола NB-Fi . . . 41
Библиография . . . 44

Введение

Настоящий стандарт предназначен для построения беспроводных сетей обмена данными между множеством конечных устройств (модемов) с одной стороны и множеством базовых станций с другой стороны. Настоящий стандарт предусматривает возможность дальнейшей интеграции данных в единое серверное пространство. Объединение всех конечных устройств (модемов) в единый сервер позволяет эффективно организовать обмен данными с различными «облачными» сервисами.

Беспроводные сети, построенные с применением стандарта NB-Fi, являются сетями класса LPWAN (Low-power Wide-area Network — энергоэффективная сеть дальнего радиуса действия), которые характеризуются высокой энергоэффективностью передачи данных и высокой емкостью сети, что позволяет использовать стандарт NB-Fi для построения телеметрических систем с большим количеством абонентов. Высокая энергоэффективность дает возможность применять в работе нелицензируемые диапазоны частот, в которых установлены ограничения на излучаемую передатчиками мощность. В основе стандарта лежит использование сверхузкополосных (Ultra Narrow Band, UNB) фазоманипулированных сигналов, которые в сочетании с помехоустойчивым кодированием позволяют достигать очень высоких значений чувствительности приема (не менее минус 150 дБм), при этом суммарная полоса частот для одновременной передачи большого количества каналов является достаточно узкой.

Сеть NB-Fi, по аналогии с мобильными сетями, использует топологию «звезда». В подобной архитектуре узловые элементы — базовые станции — должны осуществлять прием и передачу многих каналов одновременно. Для передачи множества каналов необходимо увеличение выходной мощности передатчика базовой станции. Работа в нелицензируемых диапазонах частот накладывает ограничение на выходную мощность передатчика, в том числе и для базовой станции. Поэтому для всех LPWAN-сетей концептуальной является проблема ограничения пропускной способности нисходящего канала связи.

Для приема восходящих пакетов (UPLINK-пакетов) данных со стороны базовой станции применяется принцип SDR-систем (Software-Defined Radio — программно-определяемая радиосистема), где входной радиосигнал оцифровывается во всей полосе приема и в дальнейшем подвергается программной обработке. Это позволяет выполнять демодуляцию и декодирование входных пакетов данных одновременно по всем каналам во всей полосе частот. По сути, в данной системе не существует сетки каналов, пакет данных принимается базовой станцией вне зависимости от частоты, на которой выполнена отправка. Это является ключевым свойством стандарта, позволяющим использовать недорогие генераторы частоты для формирования радиосигнала, что ранее было ограничивающим фактором при использовании UNB-сигналов. Ввиду применения простых видов модуляции UPLINK-пакеты могут быть сформированы при помощи практически любого серийного интегрального радиотрансивера. Прием UPLINK-пакетов возможен только базовой станцией. В связи с этим для реализации передачи пакетов данных в обратном, нисходящем (DOWNLINK) направлении применяются виды модуляции и скорости передачи, поддерживаемые конкретным радиотрансивером, который используется в конечных устройствах.

Настоящий стандарт подходит для телеметрических систем, в которых преобладает передача данных в восходящем направлении (от устройств к серверу). Обратный канал предназначен для передачи служебной информации сети (подтверждение доставки пакетов, регулирование скорости связи) и для отправки данных для конфигурирования и смены режимов работы устройств.

Для производства конечных устройств и базовых станций могут быть также использованы специализированные радиотрансиверы, которые позволяют использовать UPLINK-пакеты для передачи данных в обоих направлениях. При этом решается проблема ограничения пропускной способности нисходящего канала связи в LPWAN-сети — объемы данных, передаваемых в восходящем и нисходящем направлении, совпадают. Физический уровень протокола NB-Fi в этом случае реализован в самом радиотрансивере.

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

ИНТЕРНЕТ ВЕЩЕЙ

Протокол беспроводной передачи данных на основе узкополосной модуляции радиосигнала (NB-Fi)

Information technology. Internet of things. Wireless protocol based on narrow band RF modulation (NB-Fi)

Срок действия — с 2019—04—01
до 2022—04—01

1 Область применения

В настоящем стандарте установлены требования к протоколу обмена для интернета вещей в узкополосном спектре (NB-Fi), включая требования:

- к физическому уровню (раздел 5);
- MAC-уровню (раздел 6);
- транспортному уровню (раздел 7);
- уровню представления (раздел 8).

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ 14254 (МЭК 529—89) Степени защиты, обеспечиваемые оболочками (код IP)

ГОСТ Р 34.12 Информационная технология. Криптографическая защита информации. Блочные шифры

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 следующие термины с соответствующими определениями:

3.1 **радиотрансивер:** Интегральная схема, предназначенная для приема-передачи данных с использованием радиосигналов (в том числе посредством протокола NB-Fi).

3.2 **устройство приема-передачи данных/модем:** Программно-аппаратный комплекс со встроенным радиотрансивером, являющийся либо самостоятельным оборудованием, либо встроенным

Издание официальное

компонентом в конечное устройство, применяемый для приема или передачи данных с использованием радиотрансивера.

3.3 **конечное устройство:** Оборудование с устройством приема-передачи данных (модемом).

3.4 **пакет восходящего направления (UPLINK-пакет):** Пакет данных, передаваемый устройствами и принимаемый базовыми станциями.

3.5 **пакет нисходящего направления (DOWNLINK-пакет):** Пакет данных, передаваемый передатчиком базовой станции и принимаемый устройствами.

4 Сокращения

В настоящем стандарте применены следующие сокращения:

ФМн-2 — двоичная фазовая манипуляция несущей;

ОФМн-2 — относительная двоичная фазовая манипуляция несущей;

ЧМн — частотная манипуляция несущей;

«Магма» — алгоритм симметричного блочного шифрования согласно ГОСТ Р 34.12.

5 Физический уровень (Physical layer)

5.1 Общие положения

Физический уровень обеспечивает механизм приема-передачи произвольной информации по среде распространения. В данном разделе установлены требования к техническим характеристикам физического уровня для двух типов пакетов данных:

- UPLINK-пакет (см. 5.2);
- DOWNLINK-пакет (см. 5.3).

5.2 UPLINK-пакет

UPLINK-пакет представляет собой модулированные последовательности двоичных данных, сгруппированных в байты.

Описание UPLINK-пакета приведено в таблице 1. Значения параметров, приведенных в таблице 1, определены для диапазона рабочих температур от минус 40 °С до плюс 70 °С.

Таблица 1 — Основные технические характеристики UPLINK-пакета

Наименование параметра	Значение (характеристика) параметра
Полоса рабочих частот (суммарная полоса приема базовой станцией)	51,2 кГц
Скорость передачи данных, выраженная в битах	50, 400, 3200, 25 600 бит/с
Модуляция	ОФМн-2
Предельная чувствительность приема для скорости передачи данных, бит/с: - 50 - 400 - 3200 - 25 600	 Минус 150 дБм Минус 141 дБм Минус 132 дБм Минус 123 дБм
Метод разделения каналов	Частотный
Количество одновременно принимаемых каналов при скорости 50 бит/с	1024
Количество одновременно принимаемых каналов при скорости 400 бит/с	128
Количество одновременно принимаемых каналов при скорости 3200 бит/с	16

Окончание таблицы 1

Наименование параметра	Значение (характеристика) параметра
Предельная пропускная способность приема UPLINK-пакетов одной базовой станцией	20 Мбит/сут
Коэффициент ошибочных пакетов	Не более 5 %

Примечания

1 Ввиду сверхмалых значений ширины полосы сигналов используется относительная фазовая манипуляция с целью минимизации влияния ухода частоты опорного генератора за время отправки пакета. Для самой низкой скорости передачи (50 бит/с) время отправки 1 бита данных будет составлять 20 мс. Необходимую стабильность частоты обеспечивают кварцевые осцилляторы с температурной нестабильностью не более 0,5 ppm.

2 ОФМн-2 с низкой скоростью передачи битов данных может быть сформирована на аппаратном уровне не всеми радиотрансиверами. Для формирования данного вида модуляции может быть использована ЧМн с более высокой скоростью передачи битов данных.

3 Мощность тепловых шумов в полосе 50 Гц при температуре 290 К составляет:

$$N = -174 + 10 \cdot \log_{10}50 = -157 \text{ дБм}.$$

При входном коэффициенте шума базовой станции, равном 2 дБ, а также отношении сигнал/шум (Signal to Noise Ratio, SNR), равном 5 дБ, при котором достигается частота ошибок по битам (Bit Error Rate, BER) BER = 10^{-5}, вычисляют предельную теоретическую чувствительность S, равную:

$$S = -157 + 2 + 5 = -150 \text{ дБм}.$$

4 В связи с тем что на практике невозможно обеспечить привязку данных к конкретным каналам ввиду недостаточной точности задающих генераторов частоты, передачу пакетов необходимо осуществлять на псевдослучайных частотах. Возможно возникновение коллизий между различными пакетами, отправка которых пересекается по времени и частоте.

5.3 DOWNLINK-пакет

DOWNLINK-пакет представляет собой модулированные последовательности двоичных данных, сгруппированных в байты.

Описание DOWNLINK-пакета приведено в таблице 2. Значения параметров, приведенных в таблице 2, определены для диапазона рабочих температур от минус 40 °С до плюс 70 °С.

Таблица 2 — Основные технические характеристики DOWNLINK-пакета

Наименование параметра	Значение (характеристика) параметра
Полоса рабочих частот (суммарная полоса приема базовой станцией)	100 кГц
Скорость передачи данных	200, 500, 5000, 57 600 бит/с
Модуляция	ФМн-2
Предельная чувствительность приема	Минус 139 дБм
Метод разделения каналов	Частотно-временной
Предельная пропускная способность приема DOWNLINK-пакетов одной базовой станцией	10 Мбит/сут (при условии работы 100 % устройств на скорости DOWNLINK 57 600 бит/с)
Коэффициент ошибочных пакетов	Не более 5 %

Вид модуляции и скорости передачи данных выбраны таким образом, чтобы обеспечить наилучший уровень входной чувствительности при использовании серийных радиотрансиверов, присутствующих на рынке на момент написания настоящего стандарта. Предельные значения чувствительности приема подобных радиотрансиверов составляют значение порядка минус 139 дБм. При использовании скоростей 200 и 500 бит/с неточность формирования частоты задающим генератором может приводить к возникновению проблем, связанных с несовпадением частот передатчика и приемника, что вызывает потери пакетов. Для решения этой проблемы применяется алгоритм компенсации нестабильности

частот задающих генераторов. Описание алгоритма компенсации нестабильности частот задающего генератора приведено в приложении А.

5.4 Режим работы LBT (Listen Before Talk)

При включенном режиме работы LBT (Listen Before Talk — «Слушай, прежде чем сказать») устройство, прежде чем выполнить переход в режим передачи для отправки пакетов, должно переключиться в режим оценки уровня сигнала в полосе частот, в которой предполагается отправка данных. Если уровень сигнала не превышает установленного значения, что означает отсутствие передачи другим устройством, то переход в режим передачи и отправка данных могут быть выполнены. В противном случае устройство не должно выполнять передачу до тех пор, пока уровень сигнала в данной полосе частот не упадет ниже установленного значения. Для включения/выключения режима работы LBT используют конфигурационный флаг транспортного уровня FLG_LBT [см. 7.2.2.2, перечисление г)].

6 MAC-уровень (Media Access Control layer, MAC)

6.1 Общие положения

MAC-уровень обеспечивает передачу датаграммы (информационного пакета) на выбранном физическом уровне и описывает следующие параметры:

- формат полей пакета;
- способы адресации;
- методы защиты данных;
- методы контроля целостности данных;
- методы восстановления ошибок.

Описание MAC-уровня приведено в таблице 3. Значения параметров, приведенных в таблице 3, определены для диапазона рабочих температур от минус 40 °С до плюс 70 °С.

Таблица 3 — Основные технические характеристики MAC-уровня

Наименование параметра	Значение (характеристика) параметра
Номерная емкость сети	4,3 млрд устройств
Эффективные скорости передачи данных (UPLINK-пакет)	10, 80, 640, 5120 бит/с
Вид помехоустойчивого кодирования	ZIGZAG
Скорость помехоустойчивого кодирования	1/2
Длина полезных данных одного пакета	8 байт
Эффективные скорости передачи данных (DOWNLINK-пакет)	В зависимости от реализации конкретного радиотрансивера
Вид помехоустойчивого кодирования (DOWNLINK-пакет)	В зависимости от реализации конкретного радиотрансивера
Скорость помехоустойчивого кодирования	В зависимости от реализации конкретного радиотрансивера (1/2, 1/3 и т. п.)
Длина полезных данных одного пакета	От 8 до 128 байт
Алгоритм шифрования полезных данных	«Магма»
Используемый размер ключа	256 бит
Примечание — Вместо алгоритма шифрования «Магма» возможно использование другого алгоритма блочного симметричного шифрования со следующими параметрами: размер блока данных — 64 бита, размер ключа — 256 бит, при наличии достаточных оснований для этого. При этом использование других алгоритмов, не являющихся российским стандартом, может накладывать ограничения на сферу применения настоящего стандарта.	

6.2 UPLINK-пакет

Общая длина UPLINK-пакета должна составлять 40 байт. Размер поля Payload (полезные данные) должен составлять 8 байт. Соотношение эффективной скорости передачи данных к физической должно составлять 2/10. Программный код реализации функции формирования UPLINK-пакета на языке Си приведен в разделе Б.1 приложения Б.

Формат UPLINK-пакетов и рабочая полоса частот одинаковы для различных скоростей передачи данных. Порядок следования битов в байтах UPLINK-пакета — от старшего к младшему.

Структура формата UPLINK-пакета приведена в таблице 4.

Таблица 4 — Структура формата UPLINK-пакета

<table>
<tr><td colspan="4" rowspan="2">Preamble (Преамбула)</td><td colspan="4" rowspan="2">Node ID (Идентификатор, присвоенный устройству)</td><td colspan="5">Data (Данные)</td><td colspan="4">Error detection and correction (Определение ошибки и коррекция)</td></tr>
<tr><td colspan="4">Header (Заголовок)</td><td>Payload (Полезные данные)</td><td>Payload CRC (Контрольная сумма полезных данных)</td><td colspan="3">Packet CRC (Контрольная сумма пакета данных)</td><td>Zigzag code</td></tr>
<tr><td>0x97</td><td>0x15</td><td>0x7a</td><td>0x6f</td><td>ID3</td><td>ID2</td><td>ID1</td><td>ID0</td><td>7</td><td>6</td><td>5</td><td>4—0</td><td>8 байт</td><td>CRC-16</td><td>CRC-32</td><td>CRC-32</td><td>CRC-32</td><td>18 байт</td></tr>
<tr><td colspan="4"></td><td colspan="4">Zigzag source (18 байт)</td><td>SYS</td><td>ACK</td><td>MULTI</td><td>ITER</td><td colspan="5"></td></tr>
</table>

6.2.1 Поле Preamble (Преамбула)

Преамбула служит для обнаружения пакета в эфире. Алгоритм демодуляции, реализованный в базовой станции, использует данное поле для обнаружения пакета в эфире и синхронизации его последующей обработки.

6.2.2 Поле Node ID (Идентификатор, присвоенный устройству)

Поле Node ID содержит идентификатор, присвоенный устройству. Поле Node ID должно иметь размер 32 бита [для номерной емкости сети, составляющей 2^{32} (4 294 967 296) устройств]. Порядок следования байт — от старшего к младшему.

6.2.3 Поле Data (Данные)

Поле Data обрабатывается транспортным уровнем. Поле Data является составным и должно содержать два поля: поле Header (Заголовок) и поле Payload (Полезные данные).

6.2.3.1 Поле Header (Заголовок)

Поле Header является заголовком пакета. Поле Header должно иметь размер 1 байт. Поле Header должно состоять из трех системных флагов и поля ITER (см. раздел 5).

6.2.3.2 Поле Payload (Полезные данные)

Поле Payload должно иметь размер 8 байт. Данное поле может быть зашифровано с помощью блочного кода «Магма» (см. раздел 6.4).

6.2.4 Поле Payload CRC (Контрольная сумма полезных данных)

Поле Payload CRC содержит контрольную сумму незашифрованного содержимого поля Payload и поля Header. Поле Payload CRC должно иметь размер 2 байта. Поле Payload CRC используется также при вычислении частоты отправки пакета. Программный код реализации функции вычисления данного параметра на языке Си приведен в разделе Б.3 приложения Б.

6.2.5 Поле Packet CRC (Контрольная сумма пакета данных)

Поле Packet CRC содержит контрольную сумму полей Node ID, Header, Payload CRC и зашифрованного содержимого поля Payload. Используются три младших байта данного значения. Программный код реализации функции вычисления данного параметра (CRC32) на языке Си приведен в разделе Б.4 приложения Б.

6.2.6 Поле Zigzag code

Поле Zigzag code должно содержать помехоустойчивый код, вычисляемый на основании данных поля Zigzag source. Используемые таблицы данных для кодирования и декодирования, а также исходные коды программной реализации помехоустойчивого кодирования на языке Си приведены в приложении В.

6.3 DOWNLINK-пакет

Пакеты данного типа формируются при помощи аппаратной реализации используемого радиотрансивера. Внутри поля полезных данных пакета, сформированного средствами используемого радиотрансивера, должны быть размещены поле адреса, поле данных транспортного уровня и поле контрольной суммы.

Структура поля данных транспортного уровня DOWNLINK-пакета приведена в таблице 5.

Таблица 5 — Структура поля данных транспортного уровня DOWNLINK-пакета

Node ID (Идентификатор, присвоенный устройству)				Data (Данные)					Payload CRC (Контрольная сумма полезных данных)
				Header (Заголовок)				Payload (Полезные данные)	
ID3	ID1	ID0	CRC-16	7	6	5	4—0	От 8 до 128 байт, блоками по 8 байт	CRC-16
				SYS	ACK	MULTI	ITER		

6.3.1 Поле Node ID (Идентификатор, присвоенный устройству)

Поле Node ID содержит идентификатор, присвоенный устройству. Поле Node ID должно иметь размер 32 бита [для номерной емкости сети, составляющей 2^{32} (4 294 967 296) устройств]. Порядок следования байт — от старшего к младшему.

6.3.2 Поле Data (Данные)

Поле Data обрабатывается транспортным уровнем. Поле Data является составным и должно содержать два поля: поле Header (Заголовок) и поле Payload (Полезные данные).

6.3.2.1 Поле Header (Заголовок)

Поле Header является заголовком пакета. Поле Header должно иметь размер 1 байт. Поле Header должно состоять из трех системных флагов и поля ITER (см. раздел 5).

6.3.2.2 Поле Payload (Полезные данные)

Поле Payload должно иметь размер от 8 до 128 байт, кратно 8 байтам. Данное поле может быть зашифровано при помощи блочного кода «Магма» (см. раздел 6.4).

6.3.3 Поле Payload CRC (Контрольная сумма полезных данных)

Поле Payload CRC содержит контрольную сумму незашифрованного содержимого поля Payload и поля Header. Поле Payload CRC должно иметь размер 2 байта. Поле Payload CRC используется также при вычислении частоты отправки пакета. Программный код реализации функции вычисления данного параметра на языке Си приведен в разделе Б.3 приложения Б.

6.4 Шифрование поля Payload

Шифрование данных должно быть выполнено как для UPLINK-пакетов, так и для DOWNLINK-пакетов. Для шифрования следует применять алгоритм «Магма» с использованием уникального для каждого устройства ключа длиной 256 бит.

Ключ шифрования может быть записан в устройство при производстве, выделен в процессе работы устройства или отсутствовать. Значение ключа, имеющее все нули или все единицы во всех 256 битах, означает отсутствие шифрования данных.

Ключ шифрования, выделенный устройству при производстве либо в процессе работы, также сохраняется на сервере. Наличие ключа шифрования на сервере обеспечивает возможность расшифровки поля Payload.

Шифрование поля Payload для DOWNLINK-пакетов выполняется по всей длине поля блоками по 8 байт.

Механизм выделения нового ключа шифрования MAC-уровня описан в разделе Г.7. приложения Г.

7 Транспортный уровень (Transport layer)

7.1 Общие положения

Транспортный уровень обеспечивает механизмы передачи пользовательских данных и управляющих команд между базовой станцией и устройствами в условиях нестабильной связи, реализуя следующие функции:

- подтверждения доставки сообщений;
- повторной отправки данных;

- разбиения больших пакетов данных на фрагменты и последующего их «склеивания»;
- буферизации отправки данных;
- синхронизации системного времени;
- конфигурирования режимов работы;
- автоматического выбора режима работы (скорости, мощности передачи);
- смены ключа шифрования MAC-уровня.

Также на транспортном уровне реализованы алгоритмы анализа качества сигнала и выбора оптимальной скорости передачи данных. Дополнительно на транспортном уровне реализован механизм конфигурирования всех параметров функционирования устройства NB-Fi.

Ключевые особенности транспортного уровня NB-Fi, позволяющие протоколу в наибольшей степени соответствовать задачам построения LPWAN-сетей:

- низкое количество «накладных» данных, используемых для транспортного уровня;
- организация группового квитирования пакетов, позволяющего экономить использование канала связи при подтверждении приема;
- реализация специальных режимов работы для устройств с батарейным питанием.

Для передачи данных от устройства к базовой станции используются UPLINK-пакеты. Для передачи данных от базовой станции к устройству применяются DOWNLINK-пакеты либо UPLINK-пакеты для устройств, построенных на специализированных радиотрансиверах. Допускается работа в режиме передачи данных от устройства к устройству («peer-to-peer»), при этом используются DOWNLINK-пакеты либо UPLINK-пакеты для устройств, построенных на специализированных радиотрансиверах.

Реализация функций транспортного уровня предполагает буферизацию пакетов данных в виде программного стека. Минимальная глубина приемного буфера составляет 32 пакета. Это обусловлено разрядностью поля ITER (итератор), равной 5 бит, которое используется для циклической нумерации всех пакетов и их последующей идентификации при запросах повторной отправки. Таким образом, передающий узел должен хранить по крайней мере 32 последних отправленных пакета для их возможной переотправки при последующем обмене данными.

Основные режимы работы транспортного уровня и их описание приведены в таблице 6.

Таблица 6 — Основные режимы работы транспортного уровня и их описание

Режим работы	Описание
NRX (No RX)	Передача данных от устройства к серверу Устройство передает данные при необходимости, остальное время модем находится в режиме «сон». Не поддерживаются переотправка «потерянных» данных и режим автоматического выбора оптимальной скорости связи
DRX (Discontinuous RX)	Передача данных в обоих направлениях Устройство передает данные при необходимости и переходит в режим приема на непродолжительное время сразу после окончания передачи. Сервер буферизирует все запросы на отправку данных устройству и выполняет передачу данных во время «открытия» временного «окна», когда устройство переходит в режим приема. Возможна работа в режиме переотправки «потерянных» данных и режиме автоматического выбора скоростей. Данный режим используют для устройств с батарейным питанием
CRX (Continuous RX)	Передача данных в обоих направлениях Устройство передает данные при необходимости, в остальное время находится в режиме приема. Отправка данных на устройство с сервера возможна в любой момент. Все функции протокола работают в полном объеме. Возможна передача данных «peer-to-peer». Данный режим используют для устройств со стационарным питанием либо для кратковременного перехода в него с целью обмена «peer-to-peer»

7.2 Описание пакетов данных транспортного уровня

7.2.1 Формат пакета транспортного уровня в общем виде

Структура формата пакета транспортного уровня в общем виде приведена в таблице 7.

Таблица 7 — Структура формата пакета транспортного уровня в общем виде

HEADER (Заголовок)				DATA (Данные)
SYS	ACK	MULTI	ITER	
1 бит	1 бит	1 бит	5 бит	от 1 до MAXLEN байт

Описание полей:
- SYS — флаг системного пакета;
- ACK — флаг, информирующий о том, что данный пакет требует подтверждения;
- MULTI:
1) флаг групповой посылки;
2) флаг, информирующий приемную сторону о том, что за данным пакетом будет отправлен другой;
- ITER — итератор пакета;
- DATA — поле данных пакета;
- MAXLEN — максимальная длина передачи поля данных.

Пакеты транспортного уровня разделяются:
- на пользовательские пакеты;
- системные пакеты.

Системные пакеты используют для передачи служебной информации, для реализации механизмов транспортного протокола, а также для передачи полезной информации (при передаче пакетов типа GROUP и SHORT).

Полезные данные, длина которых равна параметру MAXLEN, передаются внутри пользовательского пакета.

Полезные данные длиной более чем MAXLEN передаются путем дробления пакетов и объединения их в групповую посылку. При этом первый пакет в группе является системным (GROUP), а остальные — пользовательскими.

Полезные данные, длина которых меньше параметра MAXLEN, передаются внутри системного (SHORT) пакета.

Параметр MAXLEN равен 8 для UPLINK-пакетов и может иметь значения от 8 до 128 для DOWNLINK-пакетов. Параметр MAXLEN не должен изменяться в процессе обмена данными и должен иметь одинаковые значения как у передающего, так и у приемного узла.

Структура формата пользовательского пакета приведена в таблице 8, структура формата системного пакета — в таблице 9.

Таблица 8 — Структура формата пользовательского пакета

HEADER (Заголовок)				DATA (Данные)
SYS (1 бит)	ACK (1 бит)	MULTI (1 бит)	ITER (5 бит)	PAYLOAD
0	0/1	0/1	От 0 до 31	MAXLEN байт

Таблица 9 — Структура формата системного пакета

HEADER (Заголовок)				DATA (Данные)	
SYS	ACK	MULTI	ITER	TYPE	SYS_PAYLOAD
1	0/1	0/1	От 0 до 31	1 байт	От 7 до (MAXLEN-1) байт

Описание полей:
- SYS — флаг системного пакета;
- ACK — флаг, информирующий о том, что данный пакет требует подтверждения;
- MULTI:
1) флаг групповой посылки;
2) флаг, информирующий приемную сторону о том, что за данным пакетом будет отправлен другой;

- ITER — итератор пакета;
- DATA — поле данных пакета;
- TYPE — тип системного пакета;
- SYS_PAYLOAD — полезная информация;
- MAXLEN — максимальная длина передачи поля данных.

7.2.2 Типы системных пакетов

Основные типы системных пакетов, используемые в протоколе NB-Fi, приведены в таблице 10.

Таблица 10 — Основные типы системных пакетов

CODE (код)	TYPE (тип)	SYS_PAYLOAD LENGTH, bytes (Длина полезных данных, байт)	Описание
0b1xxxxxxx	SHORT (см. 7.2.2.1)	от 7 до (MAXLEN-1)	Короткий пакет данных
0x00	ACK_P (см. 7.2.2.2)	7	Подтверждение приема
0x01	HEARTBEAT (см. 7.2.2.3)	7	Информация о параметрах работы устройства
0x02	GROUP (см. 7.2.2.5)	7	Первый пакет в группе
0x04	CLEAR (см. 7.2.2.4)	0	Сигнал завершения сеанса передачи данных
0x06	CONF (см. 7.2.2.6)	7	Настройка параметров NB-Fi
0x07	RESET (см. 7.2.2.7)	7	Сброс устройства
0x08	CLEAR_T (см. 7.2.2.78)	4	Сигнал завершения сеанса передачи данных со значением Unix Timestamp
0x09	SENDTIME (см. 7.2.2.9)	4	Сигнал синхронизации системного времени
0x0A	GETNEWKEY (см. 7.2.2.10)	0	Запрос на выделение устройством нового ключа шифрования
0x10—0x14	KEY0—KEY4 (см. 7.2.2.11)	7	Значение ключа шифрования

7.2.2.1 Пакет SHORT (короткий пакет данных)

Данный системный пакет предназначен для передачи пользовательских данных длиной менее чем MAXLEN.

Структура формата пакета SHORT приведена в таблице 11.

Таблица 11 — Структура формата пакета SHORT

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0b1xxxxxxx	SHORT	0	0x80 + LENGTH
—	—	1 — (MAXLEN-1)	PAYLOAD

Примечание — Допустимые значения LENGTH (длины полезных данных пакета): от 1 до 127 байт.

7.2.2.2 Пакет ACK_P (подтверждение приема)

Данный системный пакет предназначен для подтверждения приема пакета данных.

Структура формата пакета ACK_P приведена в таблице 12.

Таблица 12 — Структура формата пакета ACK_P

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x00	ACK_P	0	0x00
—	—	1—4	MASK [см. 7.2.2.2, а)]
—	—	5	SNR [см. 7.2.2.2, б)]
—	—	6	NOISE_OR_RTC_OFS_0_7 [см. 7.2.2.2, в)]
—	—	7	MFLAGS [см. 7.2.2.2, г)]

а) MASK

Структура формата поля MASK приведена в таблице 13.

Таблица 13 — Структура формата поля MASK

MASK0	MASK1	MASK2	MASK3
bit 0: статус пакета с итератором (*i* — 32) — bit 7: статус пакета с итератором (*i* — 25)	bit 0: статус пакета с итератором (*i* — 24) — bit 7: статус пакета с итератором (*i* — 17)	bit 0: статус пакета с итератором (*i* — 16) — bit 7: статус пакета с итератором (*i* — 9)	bit 0: статус пакета с итератором (*i* — 8) — bit 7: статус пакета с итератором (*i* — 1)

Примечание — *i*-итератор ACK_P пакета равен итератору принятого пакета, в ответ на который отправлен ACK_P.

Статус сообщения с текущим итератором *i* не включен в маску, так как факт получения пакета ACK_P подтверждает успешность доставки пакета с текущим итератором.

б) SNR (Соотношение сигнал/шум пакета)

Соотношение сигнал/шум пакета, в ответ на который отправлен ACK_P, используют для оценки качества связи при автоматическом выборе скорости и мощности передатчика. Допустимые значения — от 0 до 127 дБ.

в) NOISE_OR_RTC_OFS_0_7 (Уровень входного шума либо 8 младших бит поправки времени)

В данном поле передается уровень входного шума NOISE либо 8 младших бит поправки времени RTC_OFS_0_7. Первый вариант используется при передаче от устройства к серверу в качестве дополнительной информации; второй вариант — при передаче от сервера к устройству в случае применения механизма синхронизации времени. Допустимые значения — от 0 до 255. Данные значения соответствуют уровням шума от минус 150 до плюс 105 дБм либо 8 младшим битам 14-битной поправки времени.

г) MFLAGS

Структура формата поля MFLAGS при передаче ACK_P от сервера к устройству приведена в таблице 14.

Таблица 14 — Структура формата поля MFLAGS при передаче ACK_P от сервера к устройству

7 бит	6 бит	От 5 до 0 бит
UL_SPEED_NOT_MAX	DL_SPEED_NOT_MAX	RTC_OFS_8_13

UL_SPEED_NOT_MAX уведомляет устройство о том, что текущая скорость передачи UPLINK-пакетов является немаксимальной для данной базовой станции. Активное значение — 1.

DL_SPEED_NOT_MAX уведомляет устройство о том, что текущая скорость приема DOWNLINK-пакетов является немаксимальной для данной базовой станции. Активное значение — 1.

RTC_OFS_8_13 — старшие 6 бит 14-битной поправки времени — используют при реализации механизма синхронизации системного времени конечного устройства.

Структура формата поля MFLAGS при передаче ACK_P от устройства к серверу приведена в таблице 15.

Таблица 15 — Структура формата поля MFLAGS при передаче ACK_P от устройства к серверу

7 бит	6 бит	От 5 до 0 бит
DL_POWER_STEP_DOWN	DL_POWER_STEP_UP	TX_PWR

DL_POWER_STEP_DOWN уведомляет сервер о необходимости снизить мощность передатчика базовой станции для данного устройства. Активное значение — 1.

DL_POWER_STEP_UP уведомляет сервер о необходимости повысить мощность передатчика базовой станции для данного устройства. Активное значение — 1.

TX_PWR соответствует уровню текущей мощности передатчика устройства. Допустимые значения — от 0 до RF_MAX_POWER дБм.

7.2.2.3 Пакет HEARTBEAT (информация о параметрах работы устройства)

Данный системный пакет предназначен для передачи информации о параметрах работы устройства.

Структура формата пакета HEARTBEAT приведена в таблице 16.

Таблица 16 — Структура формата пакета HEARTBEAT

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x01	HEARTBEAT	0	0x01
—	—	1	0x00
—	—	2	VSUP [см. 7.2.2.3, а)]
—	—	3	TEMP [см. 7.2.2.3, б)]
—	—	4	AVER_RX_SNR [см. 7.2.2.3, в)]
—	—	5	AVER_TX_SNR [см. 7.2.2.3, г)]
—	—	6	NOISE [см. 7.2.2.3, д)]
—	—	7	TX_PWR [см. 7.2.2.3, е)]

а) VSUP (Напряжение питания устройства)

Должно быть представлено в следующем формате:

Старший бит: 1, если $U \geq 3$В.

Младшие 7 бит: десятые доли вольта.

Формула для перевода данного поля в вольты следующая:

$$V = 2 + (D>>7) + (D\&0x7F)/100.$$

б) TEMP (Температура внутри устройства)

Тип данных: int8_t.

Допустимые значения: от минус 128 °С до плюс 128 °С.

в) AVER_RX_SNR (Среднее значение соотношения сигнал/шум на входе приемника устройства)

Среднее значение соотношения сигнал/шум на входе приемника устройства определяют по нескольким последним принятым пакетам. Допустимые значения: от 0 до 127 дБ.

г) AVER_TX_SNR (Среднее значение соотношения сигнал/шум на входе приемника базовой станции)

Среднее значение соотношения сигнал/шум на входе приемника базовой станции определяют по нескольким последним принятым пакетам и вычисляют на основании данных, получаемых в поле SNR ACK_P пакета. Допустимые значения: от 0 до 127 дБ.

д) NOISE (Уровень шума на входе приемника устройства)

Допустимые значения: от 0 до 255. Данные значения соответствуют уровням шума от минус 150 до плюс 105 дБм.

е) TX_PWR (Уровень текущей мощности передатчика устройства)

Допустимые значения: от минус 10 до RF_MAX_POWER дБм.

7.2.2.4 Пакет CLEAR (сигнал завершения сеанса передачи данных)

Данный системный пакет предназначен для информирования о завершении сеанса передачи данных.

Структура формата пакета CLEAR приведена в таблице 17.

Таблица 17 — Структура формата пакета CLEAR

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x04	CLEAR	0	0x04
—	—	1—7	Зарезервированы

7.2.2.5 Пакет GROUP (первый пакет в группе)

Данный системный пакет предназначен для определения заголовка (первого пакета) групповой посылки.

Структура формата пакета GROUP приведена в таблице 18.

Таблица 18 — Структура формата пакета GROUP

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x02	GROUP	0	0x02
—	—	1	GROUP_LEN [см. 7.2.2.5, а)]
—	—	2	GROUP_CRC [см. 7.2.2.5, в)]
—	—	3—7	PAYLOAD_0_4 [см. 7.2.2.5, б)]

а) GROUP_LEN

Суммарное количество байт полезных данных во всей групповой посылке.

б) GROUP_CRC

Контрольная сумма CRC8 данных группы.

в) PAYLOAD_0_4

Первые 5 байт полезных данных групповой посылки.

7.2.2.6 Пакет CONF (настройка параметров NB-Fi)

Данный системный пакет предназначен для выполнения настройки параметров NB-Fi.

Структура формата пакета CONF приведена в таблице 19.

Таблица 19 — Структура формата пакета CONF

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Поле данных)
0x06	CONF	0	0x06
—	—	1	CMD&PARAM [см. 7.2.2.6, а)]
—	—	2—7	CONF_DATA

Поле CONF_DATA определено в 7.2.2.6, перечисления г)—х).

а) CMD&PARAM

Структура формата поля CMD&PARAM приведена в таблице 20.

Таблица 20 — Структура формата поля CMD&PARAM

От 7 до 6 бит	От 5 до 0 бит
CMD (см.7.2.2.6, б)	PARAM [см. 7.2.2.6, в)]

б) CMD

Структура формата поля CMD приведена в таблице 21.

Таблица 21 — Структура формата поля CMD

CODE (код)	TYPE (тип)	Описание
0x00	READ_CMD	Команда чтения параметра(ов)
0x01	WRITE_CMD	Команда записи параметра(ов)
0x02	WRITE_CMD_ACK	Команда записи параметра(ов) с верификацией и сохранением в энергонезависимой памяти
0x03	WRITE_CMD_SAVE	Команда записи параметра(ов) с сохранением в энергонезависимой памяти

в) PARAM

Структура формата поля PARAM приведена в таблице 22.

Таблица 22 — Структура формата поля PARAM

CODE (код)	TYPE (тип)	Read/Write (чтение/ запись)	Наименование параметра(ов)
0x00	NBFI_PARAM_MODE	R/W	Параметры режима работы NB-Fi
0x01	NBFI_PARAM_HANDSHAKE	R/W	Параметры режима квитирования NB-Fi
0x02	NBFI_PARAM_MAXLEN	R/W	Параметр MAXLEN
0x03	NBFI_PARAM_TXFREQ	R/W	Параметр TXFREQ (частота передачи)
0x04	NBFI_PARAM_RXFREQ	R/W	Параметр RXFREQ (частота приема)
0x05	NBFI_PARAM_ANT	R/W	Параметры RF-фронтенда (выбор антенн и мощности передачи)
0x06	NBFI_PARAM_DL_ADD	R/W	Параметр DL_ADD (адресация DOWNLINK-пакетов)
0x07	NBFI_PARAM_HEART_BEAT	R/W	Параметры отправки HEARTBEAT-пакетов
0x08	NBFI_PARAM_TX_BRATES	R	Чтение поддерживаемых скоростей передачи
0x09	NBFI_PARAM_RX_BRATES	R	Чтение поддерживаемых скоростей приема
0x0A	NBFI_PARAM_VERSION	R	Чтение версии исполнения устройства
0x0B	NBFI_ADD_FLAGS	R/W	Параметр ADDITIONAL_FLAGS
0x0C	NBFI_QUALITY	R	Чтение параметра QUALITY

Окончание таблицы 22

CODE (код)	TYPE (тип)	Read/Write (чтение/ запись)	Наименование параметра(ов)
0x0D	NBFI_UL_BASE_FREQ	R/W	Параметр UL_BASE_FREQ (базовая частота передачи)
0x0E	NBFI_DL_BASE_FREQ	R/W	Параметр DL_BASE_FREQ (базовая частота приема)
0x0F	NBFI_QUALITY_EX	R	Чтение параметра QUALITY_EXT
0x11	NBFI_APPLY_KEY	W	Команда применения нового ключа

Описание поля CONF_DATA для каждого параметра таблицы 22 приведено в 7.2.2.6, перечисления г)—х).

г) Значение поля CONF_DATA для параметра NBFI_PARAM_MODE (PARAM = 0x00)

Таблица 23 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_MODE

BYTE # (номер байта)	CONF_DATA
0	NBFI_MODE
1	NBFI_MACK_MODE
2	NBFI_TX_PHY_CHANNEL
3	NBFI_RX_PHY_CHANNEL
4	NBFI_TX_PWR
5	NBFI_NUM_OF_RETRIES

NBFI_MODE — режим работы транспортного уровня протокола NB-Fi. Описание поля NBFI_MODE приведено в таблице 24.

Таблица 24 — Описание поля NBFI_MODE

CODE (код)	TYPE (тип)
0	NRX
1	DRX
2	CRX
3	TRANSPARENT
4	OFF

NBFI_MACK_MODE — параметр группового квитирования (multiple ack) данных при отправке. Может иметь значения от 0 до 32.

NBFI_TX_PHY_CHANNEL — параметр, определяющий тип и скорость пакетов при отправке UPLINK-пакетов. Описание поля NBFI_TX_PHY_CHANNEL приведено в таблице 25.

Таблица 25 — Описание поля NBFI_TX_PHY_CHANNEL

CODE (код)	TYPE (тип)	Описание
21	UL_DBPSK_50_PROT_E	UPLINK-пакет на скорости 50 бит/с
22	UL_PSK_200	DOWNLINK-пакет на скорости 200 бит/с

Окончание таблицы 25

CODE (код)	TYPE (тип)	Описание
24	UL_DBPSK_400_PROT_E	UPLINK-пакет на скорости 400 бит/с
25	UL_PSK_500	DOWNLINK-пакет на скорости 500 бит/с
26	UL_DBPSK_3200_PROT_E	UPLINK-пакет на скорости 3200 бит/с
27	UL_PSK_5000	DOWNLINK-пакет на скорости 5000 бит/с
28	UL_DBPSK_25600_PROT_E	UPLINK-пакет на скорости 25 600 бит/с
29	UL_PSK_FASTDL	DOWNLINK-пакет на скорости 57 600 бит/с
50	UL_CARRIER	Формирование несущей частоты без модуляции

Примечания

1 DOWNLINK-пакеты применяют для отправки при взаимодействии «peer-to-peer».

2 UL_CARRIER используют для диагностических целей.

NBFI_RX_PHY_CHANNEL — параметр, определяющий тип и скорость пакетов при приеме DOWNLINK-пакетов. Описание поля NBFI_RX_PHY_CHANNEL приведено в таблице 26.

Таблица 26 — Описание поля NBFI_RX_PHY_CHANNEL

CODE (код)	TYPE (тип)	Описание
0	DL_PSK_200	DOWNLINK-пакет на скорости 200 бит/с
1	DL_PSK_500	DOWNLINK-пакет на скорости 500 бит/с
2	DL_PSK_5000	DOWNLINK-пакет на скорости 5000 бит/с
3	DL_PSK_FASTDL	DOWNLINK-пакет на скорости 57 600 бит/с

NBFI_TX_PWR — уровень выходной мощности передатчика. Тип параметра: int8_t. Допустимые значения: от минус 10 до RF_MAX_POWER дБм.

NBFI_NUM_OF_RETRIES — максимальное количество повторных отправок пакетов в течение одного сеанса отправки данных. Допустимые значения: от 0 до 255.

д) Значение поля CONF_DATA для параметра NBFI_PARAM_HANDSHAKE (PARAM = 0x01)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_HANDSHAKE приведено в таблице 27.

Таблица 27 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_HANDSHAKE

BYTE # (номер байта)	CONF_DATA
0	NBFI_HANDSHAKE_MODE
1	NBFI_MACK_MODE
2—5	Зарезервированы

NBFI_HANDSHAKE_MODE — режим квитирования данных. Описание поля NBFI_HANDSHAKE_MODE приведено в таблице 28.

Таблица 28 — Описание поля NBFI_HANDSHAKE_MODE

CODE (код)	TYPE (тип)
0	HANDSHAKE_NONE
1	HANDSHAKE_SIMPLE

NBFI_MACK_MODE — параметр группового квитирования (multiple ack) данных при отправке. Допустимые значения: от 0 до 32.

е) Значение поля CONF_DATA для параметра NBFI_PARAM_MAXLEN (PARAM = 0x02)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_MAXLEN приведено в таблице 29.

Таблица 29 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_MAXLEN

BYTE # (номер байта)	CONF_DATA
0	MAXLEN
1—5	Зарезервированы

MAXLEN — параметр, определяющий размер полезных данных одного пакета, передаваемого на MAC-уровне. Для UPLINK-пакетов данный параметр соответствует значению 8.

ж) Значение поля CONF_DATA для параметра NBFI_PARAM_TXFREQ (PARAM = 0x03)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_TXFREQ приведено в таблице 30.

Таблица 30 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_TXFREQ

BYTE # (номер байта)	CONF_DATA
0—3	TXFREQ
4—5	Зарезервированы

TXFREQ — параметр, определяющий частоту отправки данных. При TXFREQ = 0 частоту отправки данных вычисляют по формуле

$$f = f_{base_ul} + \left((ID + CRC8)\%226\right)100,$$

где f_{base_ul} — базовая частота отправки UPLINK-пакета (UL_BASE_FREQ);

ID — идентификационный номер модема;

$CRC8$ — одно из полей UPLINK-пакета, равное контрольной сумме поля PAYLOAD.

При TXFREQ = 0 чтение данного параметра возвращает значение 0 и не позволяет определить текущую частоту отправки.

При TXFREQ ≠ 0 частота отправки равна значению TXFREQ.

Порядок следования байт в данном поле данных — старшим байтом вперед.

и) Значение поля CONF_DATA для параметра NBFI_PARAM_RXFREQ (PARAM = 0x04)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_RXFREQ приведено в таблице 31.

Таблица 31 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_RXFREQ

BYTE # (номер байта)	CONF_DATA
0—3	RXFREQ
4—5	Зарезервированы

RXFREQ — параметр, определяющий частоту, на которой выполняется прием данных. При RXFREQ = 0 частоту *f* вычисляют по формуле

$$f = f_{base_dl} + (ID\%276)363,$$

где f_{base_dl} — базовая частота приема DL (DL_BASE_FREQ);

ID — идентификационный номер модема.

При RXFREQ = 0 чтение данного параметра возвращает значение 0 и не позволяет определить текущую частоту приема.

При RXFREQ ≠ 0 частота приема равна значению RXFREQ.

Порядок следования байтов в данном поле данных — старшим байтом вперед.

к) Значение поля CONF_DATA для параметра NBFI_PARAM_ANT (PARAM = 0x05)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_ ANT приведено в таблице 32.

Таблица 32 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_ANT

BYTE # (номер байта)	CONF_DATA
0	NBFI_TX_PWR
1	NBFI_TX_ANT
2	NBFI_RX_ANT
3—5	Зарезервированы

NBFI_TX_PWR — уровень выходной мощности передатчика. Тип параметра: int8_t. Допустимые значения: от минус 10 до RF_MAX_POWER дБм.

NBFI_TX_ANT — параметр, определяющий, какой из RF-трактов используется при передаче. Значения параметра и соответствующие им варианты RF-трактов зависят от конкретной реализации устройства.

NBFI_RX_ANT — параметр, определяющий, какой из RF-трактов используется при приеме. Значения параметра и соответствующие им варианты RF-трактов зависят от конкретной реализации устройства.

л) Значение поля CONF_DATA для параметра NBFI_PARAM_DL_ADD (PARAM = 0x06)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_ DL_ADD приведено в таблице 33.

Таблица 33 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_DL_ADD

BYTE # (номер байта)	CONF_DATA
0—2	NBFI_DL_ADD
3—5	Зарезервированы

NBFI_DL_ADD — параметр, использующийся для адресации при отправке и приеме DOWNLINK-пакетов. По умолчанию равен ID-адресу устройства. Может быть изменен на адрес другого устройства для приема пакетов только от заданного узла.

м) Значение поля CONF_DATA для параметра NBFI_PARAM_HEART_BEAT (PARAM = 0x07)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_ HEART_BEAT приведено в таблице 34.

Таблица 34 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_HEART_BEAT

BYTE # (номер байта)	CONF_DATA
0	NBFI_HEARTBEAT_NUM
1—2	NBFI_HEARTBEAT_INTERVAL
3—5	Зарезервированы

NBFI_HEARTBEAT_NUM — параметр, определяющий количество HEARTBEAT-пакетов, которые будут отправлены после сброса устройства. При NBFI_HEARTBEAT_NUM = 255 отправка HEARTBEAT-пакетов выполняется неограниченное количество раз.

NBFI_HEARTBEAT_INTERVAL — параметр, определяющий периодичность отправки HEARTBEAT-пакетов. Допустимые значения: от 0 до 65 535. Порядок следования байтов в поле параметра: старшим байтом вперед. Порядок расчета периодичности отправки HEARTBEAT-пакетов при разных режимах работы транспортного уровня протокола NB-Fi (NBFI_MODE) приведен в таблице 35.

Таблица 35 — Порядок расчета периодичности отправки HEARTBEAT-пакетов при разных режимах работы транспортного уровня протокола NB-Fi

NBFI_MODE	Периодичность отправки HEARTBEAT-пакетов, с
NRX, DRX	NBFI_HEARTBEAT_INTERVAL * 60
CRX	NBFI_HEARTBEAT_INTERVAL

н) Значение поля CONF_DATA для параметра NBFI_PARAM_TX_BRATES (PARAM = 0x08)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_ TX_BRATES приведено в таблице 36.

Таблица 36 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_TX_BRATES

BYTE # (номер байта)	CONF_DATA
0—5	NBFI_TX_BRATES

NBFI_TX_BRATES — в данном поле перечислены все поддерживаемые устройством скорости передачи данных, которые используются при автоматическом выборе оптимальной скорости. Данные представлены в формате значений параметра NBFI_TX_PHY_CHANNEL [см. 7.2.2.6, г)]. Максимально возможное количество поддерживаемых скоростей равно шести. Поддерживаемые скорости перечислены начиная с младшего байта данного поля. Если число скоростей менее шести, старшие (неиспользуемые) байты данного поля равны 0xFF.

Примечание — Параметр NBFI_PARAM_TX_BRATES доступен только для чтения.

п) Значение поля CONF_DATA для параметра NBFI_PARAM_RX_BRATES (PARAM = 0x09)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_ RX _BRATES приведено в таблице 37.

Таблица 37 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_RX_BRATES

BYTE # (номер байта)	CONF_DATA
0—5	NBFI_RX_BRATES

NBFI_RX_BRATES — в данном поле перечислены все поддерживаемые устройством скорости приема данных, которые используются при автоматическом выборе оптимальной скорости. Данные представлены в формате значений параметра NBFI_RX_PHY_CHANNEL [см. 7.2.2.6, г)]. Максимально возможное количество поддерживаемых скоростей равно шести. Поддерживаемые скорости перечислены начиная с младшего байта данного поля. Если число скоростей менее шести, старшие (неиспользуемые) байты данного поля равны 0xFF.

Примечание — Параметр NBFI_PARAM_RX_BRATES доступен только для чтения.

р) Значение поля CONF_DATA для параметра NBFI_PARAM_VERSION (PARAM = 0x0A)

Описание структуры формата поля CONF_DATA для параметра NBFI_PARAM_ VERSION приведено в таблице 38.

Таблица 38 — Структура формата поля CONF_DATA для параметра NBFI_PARAM_VERSION

BYTE # (номер байта)	CONF_DATA
0	NBFI_REV
1	NBFI_SUBREV
2	COMP_CRC
3	HARDWARE_ID
4	HARDWARE_REV
5	BAND_ID

NBFI_REV — версия транспортного протокола NB-Fi. Текущее значение равно 2.

NBFI_SUBREV — дополнительная версия транспортного протокола NB-Fi — в данный момент не используется.

COMP_CRC — контрольная сумма *CRC8* даты и времени компиляции программного обеспечения устройства — используется для автоматического контроля модификаций программного обеспечения.

HARDWARE_ID — идентификатор аппаратной платформы устройства.

HARDWARE_REV — версия аппаратной платформы устройства.

BAND_ID — идентификатор радиочастотного решения устройства — описывает рабочие частоты приемного и передающего трактов.

Примечание — Параметр NBFI_PARAM_VERSION доступен только для чтения.

с) Значение поля CONF_DATA для параметра NBFI_ADD_FLAGS (PARAM = 0x0B)

Описание структуры формата поля CONF_DATA для параметра NBFI_ ADD_FLAGS приведено в таблице 39.

Таблица 39 — Структура формата поля CONF_DATA для параметра NBFI_ADD_FLAGS

BYTE # (номер байта)	CONF_DATA
0	NBFI_ADDITIONAL_FLAGS
1—5	Зарезервированы

NBFI_ADDITIONAL_FLAGS — параметр, представляющий собой совокупность дополнительных флагов, управляющих определенными функциями протокола, сгруппированных в виде битовой маски. Активное значение каждого флага — 1.

Описание поля NBFI_ADDITIONAL_FLAGS приведено в таблице 40.

Таблица 40 — Описание поля NBFI_ADDITIONAL_FLAGS

BIT # (номер бита)	FLAG (дополнительные флаги)
0(LSB)	FLG_FIXED_BAUD_RATE
1	FLG_NO_RESET_TO_DEFAULTS
2	FLG_NO_SENDINFO
3	FLG_NO_ENC
4	FLG_DO_OSCCAL
5	FLG_LBT
6	NBFI_OFF_MODE_ON_INIT
7(MSB)	NBFI_FLG_DO_NOT_SEND_PKTS_ON_START

FLG_FIXED_BAUD_RATE — данный флаг деактивирует механизм автоматического выбора оптимальной скорости и мощности передачи.

FLG_NO_RESET_TO_DEFAULTS — данный флаг деактивирует функцию сброса режима скоростей к значениям по умолчанию после неудачного выполнения сеанса передачи данных и используется совместно с FLG_FIXED_BAUD_RATE — флагом при соединениях «peer-to-peer».

FLG_NO_SEND_INFO — данный флаг отключает автоматическую отправку информационных пакетов.

FLG_NO_ENC — данный флаг отключает шифрование данных как для передачи, так и для приема пакетов. Значение данного флага влияет только на поведение устройства и не отключает функции шифрования на сервере. Используется при соединениях «peer-to-peer».

FLG_DO_OSCCAL — данный флаг активирует механизм периодической калибровки внутреннего RC-генератора. В качестве опорного генератора применяется радиоосциллятор и используется для некоторых решений на радиотрансивере AX8052F143.

FLG_LBT — данный флаг активирует режим передачи LBT (listen before talk).

NBFI_OFF_MODE_ON_INIT — данный флаг сигнализирует устройству переходить в режим работы OFF при инициализации устройства.

NBFI_FLG_DO_NOT_SEND_PKTS_ON_START — данный флаг отключает отправку системных пакетов CLEAR и CONF при инициализации устройства.

т) Значение поля CONF_DATA для параметра NBFI_QUALITY (PARAM = 0x0C)

Описание структуры формата поля CONF_DATA для параметра NBFI_QUALITY приведено в таблице 41.

Таблица 41 — Структура формата поля CONF_DATA для параметра NBFI_QUALITY

BYTE # (номер байта)	CONF_DATA
0—1	UL_TOTAL
2—3	DL_TOTAL
4	AVER_RX_SNR
5	AVER_TX_SNR

UL_TOTAL — общее количество отправленных пакетов с момента сброса устройства. Тип данного параметра: uint16_t. Порядок следования байтов в данном поле — старшим байтом вперед.

DL_TOTAL — общее количество принятых пакетов с момента сброса устройства. Тип данного параметра: uint16_t. Порядок следования байтов в данном поле — старшим байтом вперед.

AVER_RX_SNR — среднее значение соотношения сигнал/шум на входе приемника устройства, определяемое по нескольким последним принятым пакетам. Допустимые значения: от 0 до 127 дБ.

AVER_TX_SNR — среднее значение соотношения сигнал/шум на входе приемника базовой станции, определяемое по нескольким последним принятым пакетам, вычисляется на основании данных, получаемых в поле SNR ACK_P пакета. Допустимые значения: от 0 до 127 дБ.

Примечания

1 Параметр NBFI_QUALITY доступен только для чтения.

2 Параметры DL_TOTAL, AVER_RX_SNR, AVER_TX_SNR актуальны только для режимов работы DRX и CRX с NBFI_HANDSHAKE_MODE = HANDSHAKE_SIMPLE.

у) Значение поля CONF_DATA для параметра NBFI_UL_BASE_FREQ (PARAM = 0x0D)

Описание структуры формата поля CONF_DATA для параметра NBFI_UL_BASE_FREQ приведено в таблице 42.

Таблица 42 — Структура формата поля CONF_DATA для параметра NBFI_UL_BASE_FREQ

BYTE # (номер байта)	CONF_DATA
0—3	UL_BASE_FREQ
4—5	Зарезервированы

UL_BASE_FREQ — параметр соответствует базовой частоте, используемой при расчете частоты, на которой отправляются данные.

ф) Значение поля CONF_DATA для параметра NBFI_DL_BASE_FREQ (PARAM = 0x0E)

Описание структуры формата поля CONF_DATA для параметра NBFI_ DL_BASE_FREQ приведено в таблице 43.

Таблица 43 — Структура формата поля CONF_DATA для параметра NBFI_DL_BASE_FREQ

BYTE # (номер байта)	CONF_DATA
0—3	DL_BASE_FREQ
4—5	Зарезервированы

DL_BASE_FREQ — параметр соответствует базовой частоте, используемой при расчете частоты, на которой принимаются данные.

х) Значение поля CONF_DATA для параметра NBFI_QUALITY_EX (PARAM = 0x0F)

Описание структуры формата поля CONF_DATA для параметра NBFI_ QUALITY_EX приведено в таблице 44.

Таблица 44 — Структура формата поля CONF_DATA для параметра NBFI_QUALITY_EX

BYTE # (номер байта)	CONF_DATA
0	UL_RATING
1	DL_RATING
2—3	UL_SUCCESS_TOTAL
4—5	UL_FAULT_TOTAL

UL_RATING — показатель уровня сигнала на входе приемника, вычисляется на основании текущего режима скорости и среднего значения соотношения сигнал/шум на входе приемника. Допустимые значения: от 0 до 10.

DL_RATING — показатель уровня сигнала от данного устройства на входе приемника базовой станции, вычисляется на основании текущего режима скорости и среднего значения соотношения сигнал/шум на входе базовой станции. Допустимые значения: от 0 до 10.

UL_SUCCESS_TOTAL — общее количество успешно доставленных пакетов с момента сброса устройства. Тип данного параметра: uint16_t. Порядок следования байтов в данном поле — старшим байтом вперед. Для режима работы NRX либо в случае NBFI_HANDSHAKE_MODE = HANDSHAKE_NONE данный параметр равен общему количеству отправленных пакетов.

UL_FAULT_TOTAL — общее количество недоставленных пакетов с момента сброса устройства. Тип данного параметра: uint16_t. Порядок следования байтов в данном поле: старшим байтом вперед. Пакет считается недоставленным, если его не удалось доставить в результате сеанса передачи данных.

Примечания

1 Параметр NBFI_QUALITY_EX доступен только для чтения.

2 Параметры UL_RATING, DL_RATING, UL_FAULT_TOTAL актуальны только для режимов работы DRX и CRX с NBFI_HANDSHAKE_MODE = HANDSHAKE_SIMPLE.

ц) Значение поля CONF_DATA для параметра NBFI_APPLY_KEY (PARAM = 0x11)

Описание структуры формата поля CONF_DATA для параметра NBFI_ APPLY_KEY приведено в таблице 45.

Таблица 45 — Структура формата поля CONF_DATA для параметра NBFI_APPLY_KEY

BYTE # (номер байта)	CONF_DATA
0—3	KEY_CRC32
4—5	Зарезервированы

KEY_CRC32 — контрольная сумма ключа шифрования. Программный код реализации функции вычисления данного параметра (CRC32) на языке Си приведен в разделе Б.4 приложения Б.

7.2.2.7 Пакет RESET (сброс устройства)

Данный системный пакет предназначен для выполнения процедуры программного перезапуска устройства.

Структура формата пакета RESET приведена в таблице 46.

Таблица 46 — Структура формата пакета RESET

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x07	RESET	0	0x07
—	—	1	0xDE
—	—	2	0xAD
—	—	3—7	Зарезервированы

7.2.2.8 Пакет CLEAR_T. Сигнал завершения сеанса передачи данных со значением Unix Timestamp

Структура формата пакета CLEAR_T приведена в таблице 47.

Таблица 47 — Структура формата пакета CLEAR_T

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x08	CLEAR_T	0	0x08
—	—	1—4	UTS
—	—	5—7	Зарезервированы

UTS — 4-байтное значение системного времени в формате Unix Timestamp (количество секунд от 1 января 1970 года), соответствующее часовому поясу UTC.

7.2.2.9 Пакет SENDTIME. Сигнал синхронизации системного времени

Структура формата пакета SENDTIME приведена в таблице 48.

Таблица 48 — Структура формата пакета SENDTIME

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x09	SENDTIME	0	0x09
—	—	1—4	UTS
—	—	5—7	Зарезервированы

UTS — 4-байтное значение системного времени в формате Unix Timestamp (количество секунд от 1 января 1970 г.), соответствующее часовому поясу UTC.

7.2.2.10 Пакет GETNEWKEY. Запрос на выделение устройством нового ключа шифрования

Структура формата пакета GETNEWKEY приведена в таблице 49.

Таблица 49 — Структура формата пакета GETNEWKEY

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x0A	GETNEWKEY	0	0x0A
—	—	1—7	Зарезервированы

7.2.2.11 Пакет KEYn. Значение ключа шифрования

Структура формата пакета KEYn приведена в таблице 50.

Таблица 50 — Структура формата пакета KEYn

CODE (код)	TYPE (тип)	BYTE # (номер байта)	DATA (Данные)
0x1n	KEYn	0	0x1n
—	—	1—7 (3, для n = 4)	7 байт ключа шифрования(4 байта, для n = 4)
—	—	4—7	Зарезервировано для n = 4

Примечание — Параметр n принимает значения от 0 до 4.

8 Уровень представления (Presentation layer)

8.1 Общие положения

На данном уровне реализованы механизмы, обеспечивающие совместимость устройств на уровне приложения, а также обеспечивающие защиту данных уровня приложений, идентификацию устройств и чтение/запись типов данных специального назначения.

Пакет данных уровня представления передается в обоих направлениях внутри поля Payload транспортного уровня.

Описанные ниже форматы пакетов представляют собой соглашения, выполнение которых обязательно требуется в системах, к которым предъявляются требования в части безопасности передачи данных. В этих системах должен быть использован алгоритм шифрования «Магма».

При наличии достаточных оснований на использование другого алгоритма шифрования или в зависимости от требований безопасности в стране применения стандарта возможен выбор другого алгоритма симметричного блочного шифрования с размером ключа шифрования 256 бит.

8.2 Описание пакетов данных уровня представления

8.2.1 Формат пакета уровня представления

Структура формата пакета уровня представления в общем виде приведена в таблице 51.

Таблица 51 — Структура формата пакета уровня представления в общем виде

Поле	Размер
0xE0	1 байт
CMD	1 байт
ITER	2 байта
ENC	2 бита
TYPE	6 бит
CRC16	2 байта
DATA	1—233 байта

Описание полей:

- 0xE0: признак пакета уровня представления;
- CMD: код команды, который определяет назначение пакета; типы команд приведены в таблице 52;
- ITER: итератор пакета — 16-битный параметр, который должен увеличиваться на 1 для каждого нового пакета, позволяет идентифицировать уникальность пакета и реализовать защиту от атаки повторного воспроизведения;
- ENC: тип используемого шифрования; типы шифрования приведены в таблице 53;
- TYPE: тип данных пакета, который определяет тип содержимого поля DATA; типы данных приведены в таблице 54; программный код реализации функции вычисления данного параметра на языке Си приведен в разделе Б.3 приложения Б;
- DATA: поле полезных данных пакета, которое может быть зашифровано одним из видов блочного шифрования, приведенных в таблице 53.

8.2.2 Типы команд уровня представления

Таблица 52 — Типы команд уровня представления

CODE (код)	CMD (команда)	Наименование команды
0x00	NBFI_PL_GET_MAN_ID	Получение идентификатора производителя
0x01	NBFI_PL_GET_HW_ID	Получение идентификатора устройства
0x02	NBFI_PL_GET_PROT_ID	Получение идентификатора протокола устройства
0x03—0x1F	Зарезервировано	
0x20	NBFI_PL_ENC_SET_NONE	Запрос на выключение шифрования
0x21	NBFI_PL_ENC_SET_MAGMA	Запрос на включение алгоритма шифрования «Магма»
0x22	NBFI_PL_ENC_TYPE_CUSTOMCRYPTO1	Запрос на включение пользовательского шифрования № 1
0x23	NBFI_PL_ENC_TYPE_CUSTOMCRYPTO2	Запрос на включение пользовательского шифрования № 2
0x24	NBFI_PL_ENC_NEW_KEY	Запрос на смену ключа шифрования

Окончание таблицы 52

CODE (код)	CMD (команда)	Наименование команды
0x25	NBFI_PL_ENC_APPLY_KEY	Команда подтверждения смены ключа шифрования
0x26—0x3F	Зарезервировано	
0x40—0xFF	Свободно для использования приложением	

8.2.3 Типы шифрования уровня представления

Таблица 53 — Типы шифрования уровня представления

CODE (код)	ENC	Тип шифрования
0b00	NBFI_PL_ENC_TYPE_NONE	Шифрование отсутствует
0b01	NBFI_PL_ENC_TYPE_MAGMA	«Магма»
0b10	NBFI_PL_ENC_TYPE_CUSTOMCRYPTO1	Другой тип шифрования (пользовательский тип шифрования № 1)
0b11	NBFI_PL_ENC_TYPE_CUSTOMCRYPTO2	Другой тип шифрования (пользовательский тип шифрования № 2)

8.2.4 Типы данных уровня представления

Таблица 54 — Типы данных уровня представления

CODE (код)	TYPE (тип)	Наименование типа
0x00 — 0x3F	Свободно для использования приложением	

Приложение А
(справочное)

Описание алгоритма компенсации нестабильности частот задающего генератора

Суть алгоритма компенсации частот заключается в следующем:

Отправка UPLINK-пакетов устройствами осуществляется на известной (предполагаемой) частоте f_{exp_ul}, вычисляемой по формуле

$$f_{exp_ul} = f_{base_ul} + \left((ID + CRC8)\%226\right)100, \quad (A.1)$$

где f_{base_ul} — граничная частота диапазона частот, на которых выполняется отправка UPLINK-пакетов;

ID — идентификационный номер модема;

$CRC8$ — одно из полей UPLINK-пакета, равное контрольной сумме поля PAYLOAD.

Отправка DOWNLINK-пакетов базовой станцией осуществляется на известной (предполагаемой) частоте f_{exp_dl}, вычисляемой по формуле

$$f_{exp_dl} = f_{base_dl} + (ID\%276)363, \quad (A.2)$$

где f_{base_dl} — граничная частота диапазона частот, на которых выполняется отправка DOWNLINK-пакетов;

ID — идентификационный номер модема.

Базовая станция, принимая пакет, вычисляет частоту приема f_{rx1} с разрешением в несколько десятков герц (для скорости 50 бит/с) и затем выполняет вычисление обобщенной ошибки частоты Δf_1 по формуле

$$\Delta f_1 = f_{exp_ul} - f_{rx1}. \quad (A.3)$$

В состав данной ошибки входят как погрешность передатчика удаленного модема Δf_{modTX}, так и ошибка измерения частоты базовой станцией Δf_{bsRX}, таким образом

$$\Delta f_1 = \Delta f_{modTX} + \Delta f_{bsRX}. \quad (A.4)$$

Данные погрешности имеют достаточно стабильные значения, вызванные начальной неточностью генераторов, и при использовании термокомпенсированных осцилляторов незначительно изменяются при колебаниях температуры.

Передатчик базовой станции, используемый для отправки DOWNLINK-пакетов, также с определенной периодичностью (например, каждые 5 мин) отправляет UPLINK-пакеты на фиксированной частоте f_{test_dl}, которая попадает в полосу частот приема базовой станции. Базовая станция, принимая данный пакет, вычисляет частоту приема f_{rx2} и затем значение обобщенной ошибки частоты Δf_2 по формуле

$$\Delta f_2 = f_{test_dl} - f_{rx2}. \quad (A.5)$$

В состав данной ошибки входят как погрешность передатчика базовой станции Δf_{bsTX}, так и ошибка измерения частоты базовой станцией Δf_{bsRX}, таким образом

$$\Delta f_2 = \Delta f_{bsTX} + \Delta f_{bsRX}. \quad (A.6)$$

При отправке DOWNLINK-пакетов каждый раз вносится поправка в частоту передачи базовой станции прибавлением обобщенной ошибки Δf_1 и вычитанием обобщенной ошибки Δf_2 по формуле

$$f_{tx} = f_{exp_dl} + \Delta f_1 - \Delta f_2. \quad (A.7)$$

Подставляя в формулу (А.7) выражения (А.4) и (А.6), а также добавляя ошибку передатчика базовой станции Δf_{bsTX}, вычисляют фактическое значение частоты отправки f_{factTX} по формуле

$$f_{factTX} = f_{exp_dl} + \Delta f_{modTX} + \Delta f_{bsRX} - \Delta f_{bsTX} - \Delta f_{bsRX} + \Delta f_{bsTX} = f_{exp_dl} + \Delta f_{modTX}. \quad (A.8)$$

Для успешного приема сообщения данная частота f_{factTX} должна соответствовать фактической частоте приема модема f_{factRX}, которая равна:

$$\Delta f_{factRX} = f_{exp_dl} + \Delta f_{modRX}. \quad (A.9)$$

Когда частота приема и частота передачи модема совпадают либо имеют близкие значения, ошибки приема и передачи также приблизительно равны:

$$\Delta f_{modTX} \approx \Delta f_{modRX}. \quad (A.10)$$

Таким образом, по формуле (А.7) вычисляют необходимую компенсацию ошибок генераторов для случая

$$f_{base_ul} \approx f_{base_dl}. \quad (A.11)$$

В общем случае формула компенсации погрешностей частот имеет следующий вид:

$$f_{tx} = f_{exp_dl} + \frac{f_{base_dl}}{f_{base_ul}}\left(\Delta f_1 - \Delta f_2\right). \quad (A.12)$$

Подставив в формулу (А.12) формулы (А.3) и (А.5), получаем полную формулу расчета частоты отправки DOWNLINK-пакетов, при которой выполняется коррекция ошибок всех генераторов:

$$f_{tx} = f_{exp_dl} + \frac{f_{base_dl}}{f_{base_ul}}\left(f_{exp_ul} - f_{rx1} - f_{\mathrm{test}_{dl}} + f_{rx2}\right). \quad (A.13)$$

Приложение Б
(справочное)

Фрагменты исходных кодов реализации MAC-уровня

Б.1 Функция формирования и отправки UPLINK-пакета

```
nbfi_status_t NBFi_TX_ProtocolE(nbfi_transport_packet_t* pkt)
{
    uint8_t ul_buf[64];
    uint8_t len = 0;
    static _Bool parity = 0;
    uint16_t lastcrc16;
    memset_xdata(ul_buf,0,sizeof(ul_buf));
    if(nbfi.mode == TRANSPARENT) pkt->phy_data_length--;
    for(int i=0; i<sizeof(protE_preambula); i++)
    {
        ul_buf[len++] = protE_preambula[i];
    }
    ul_buf[len++] = nbfi.full_ID[0];
    ul_buf[len++] = nbfi.full_ID[1];
    ul_buf[len++] = nbfi.full_ID[2];
    ul_buf[len++] = nbfi.full_ID[3];
    if(nbfi.tx_phy_channel == DL_DBPSK_50_PROT_E) nbfi.tx_phy_channel =
UL_DBPSK_50_PROT_E;
    else if(nbfi.tx_phy_channel == DL_DBPSK_400_PROT_E) nbfi.tx_phy_channel =
UL_DBPSK_400_PROT_E;
    else if(nbfi.tx_phy_channel == DL_DBPSK_3200_PROT_E) nbfi.tx_phy_channel =
UL_DBPSK_3200_PROT_E;
    else if(nbfi.tx_phy_channel == DL_DBPSK_25600_PROT_E) nbfi.tx_phy_channel =
UL_DBPSK_25600_PROT_E;
    ul_buf[len++] = pkt->phy_data.header;
    memcpy_xdatageneric(&ul_buf[len], pkt->phy_data.payload, pkt->phy_data_length);
    lastcrc16 = CRC16(&ul_buf[len - 1], 9, 0xFFFF);
    if(MAGMA_Enabled() && MAGMA _Available() && !(nbfi.additional_flags&NBFI_FLG_NO_
MAGMA))
    {
        MAGMA _Encode(&ul_buf[len]);
    }
    len += 8;
    if(nbfi.mode == TRANSPARENT)
    {
        ul_buf[len++] = pkt->phy_data.payload[8];
        ul_buf[len++] = pkt->phy_data.payload[9];
    }
    else
    {
        ul_buf[len++] = lastcrc16&0xff;
        ul_buf[len++] = (lastcrc16>>8)&0xff;
    }
    last_pkt_crc = CRC32(ul_buf + 4, 15);

    ul_buf[len++] = (uint8_t)(last_pkt_crc >> 16);
    ul_buf[len++] = (uint8_t)(last_pkt_crc >> 8);
    ul_buf[len++] = (uint8_t)(last_pkt_crc);
    if(nbfi.tx_freq)
    {
       tx_freq = nbfi.tx_freq ;
       parity = (nbfi.tx_freq > (nbfi.ul_freq_base + 25000));
    }
```

```
    else
    {
      if(nbfi.tx_phy_channel < UL_DBPSK_3200_PROT_E)
      {
         tx_freq = nbfi.ul_freq_base + (((*((const uint32_t *)FULL_
ID)+lastcrc16)%226)*100);
        if(parity) tx_freq = tx_freq + 27500;
      }
      else
      {
        tx_freq = nbfi.ul_freq_base + 1600 + (((*((const uint32_t *)FULL_
ID)+lastcrc16)%210)*100);
        if(parity) tx_freq = tx_freq + 27500 - 1600;
      }
    }
    ZCODE_E_Append(&ul_buf[4], &ul_buf[len], 1);
    if(!nbfi.tx_freq) parity = !parity;
    if((nbfi.mode == NRX) && parity)
    {
        RF_Init(nbfi.tx_phy_channel, (rf_antenna_t)nbfi.tx_antenna, nbfi.tx_pwr, tx_
freq);
        RF_Transmit(ul_buf, len + ZCODE_LEN, PADDING_4TO1, BLOCKING);
        nbfi_state.UL_total++;
        return NBFi_TX_ProtocolE(pkt);
    }
    RF_Init(nbfi.tx_phy_channel, (rf_antenna_t)nbfi.tx_antenna, nbfi.tx_pwr, tx_freq);
    RF_Transmit(ul_buf, len + ZCODE_LEN, PADDING_4TO1, NONBLOCKING);
    nbfi_state.UL_total++;
    return OK;
}
```

Б.2 Функция вычисления контрольной суммы (CRC8)

```
static unsigned char CRC8byte(unsigned char data)
{
   uint8_t crc = 0;
   if(data & 1)      crc ^= 0x5e;
   if(data & 2)      crc ^= 0xbc;
   if(data & 4)      crc ^= 0x61;
   if(data & 8)      crc ^= 0xc2;
   if(data & 0x10)   crc ^= 0x9d;
   if(data & 0x20)   crc ^= 0x23;
   if(data & 0x40)   crc ^= 0x46;
   if(data & 0x80)   crc ^= 0x8c;
   return crc;
}

uint8_t CRC8(uint8_t* data, uint8_t len)
{
    uint8_t crc = 0;
    for(uint8_t i = 0; i < len; i++)
    {
        crc = CRC8byte(data[i] ^ crc);
    }
    return crc;
}
```

Б.3 Функция вычисления контрольной суммы (CRC16)

```
#define POLY 0xa001
uint16_t CRC16(uint8_t *buf, uint16_t len, uint16_t crc)
{
```

```
    while (len--)
    {
        crc ^= *buf++;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
        crc = crc & 1 ? (crc >> 1) ^ POLY : crc >> 1;
    }
    return crc;
}
```

Б.4 Функция вычисления контрольной суммы (CRC32)

```
uint32_t CRC32(const uint8_t *buf, uint8_t len)
{
    return digital_update_crc32(0xffffffff, buf, len) ^ 0xffffffff;
}

#define WIDTH (8*4)
#define TOPBIT (1 << (WIDTH-1))
#define POLYNOMIAL (0x104C11DB7)

uint32_t crc_table(uint8_t n)
{
    uint32_t c;
    int k;
    c=((uint32_t)n) << (WIDTH - 8);
    for(k=8;k>0;k--)
    {
         if(c & (uint32_t)TOPBIT)
        {
            c = (c<<1) ^ POLYNOMIAL;
        }
        else
        {
            c=c<<1;
        }
    }
    return c;
}

uint32_t digital_update_crc32( uint32_t crc, const uint8_t *data, uint8_t len )
{
    while (len > 0)
    {
     crc = crc_table(*data ^ ((crc >> 24) & 0xff)) ^ (crc << 8);
     data++;
     len--;
    }
    return crc;
}
```

Приложение В
(справочное)

Таблица и функции, используемые для ZIGZAG-кодирования данных

```
// Zigzag code permutation table
const uint8_t n_e[4][144] =
{


{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,
63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,
93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,
117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,
139,140,141,142,143},
{128,96,64,32,0,113,81,49,17,130,98,66,34,2,115,83,51,19,132,100,68,36,4,117,85,53,21,134,
102,70,38,6,119,87,55,23,136,104,72,40,8,121,89,57,25,138,106,74,42,10,123,91,59,27,140,
108,76,44,12,125,93,61,29,142,110,78,46,14,127,95,63,31,112,80,48,16,129,97,65,33,1,114,
82,50,18,131,99,67,35,3,116,84,52,20,133,101,69,37,5,118,86,54,22,135,103,71,39,7,120,88,
56,24,137,105,73,41,9,122,90,58,26,139,107,75,43,11,124,92,60,28,141,109,77,45,13,126,94,
62,30,143,111,79,47,15},
{93,136,34,77,120,18,61,104,2,45,88,131,29,72,115,13,56,99,142,
40,83,126,24,67,110,8,51,94,137,35,78,121,19,62,105,3,46,89,132,30,73,116,14,57,100,143,
41,84,127,25,68,111,9,52,95,138,36,79,122,20,63,106,4,47,90,133,31,74,117,15,58,101,42,
85,128,26,69,112,10,53,96,139,37,80,123,21,64,107,5,48,91,134,32,75,118,16,59,102,0,43,86,
129,27,70,113,11,54,97,140,38,81,124,22,65,108,6,49,92,135,33,76,119,17,60,103,1,44,87,130,
28,71,114,12,55,98,141,39,82,125,23,66,109,7,50},
{2,8,14,20,26,32,38,44,50,56,62,68,74,80,86,92,98,104,110,116,122,128,134,140,1,7,13,19,
25,31,37,43,49,55,61,67,73,79,85,91,97,103,109,115,121,127,133,139,0,6,12,18,24,30,36,42,
48,54,60,66,72,78,84,90,96,102,108,114,120,126,132,138,5,11,17,23,29,35,41,47,53,59,65,71,
77,83,89,95,101,107,113,119,125,131,137,143,4,10,16,22,28,34,40,46,52,58,64,70,76,82,88,
94,100,106,112,118,124,130,136,142,3,9,15,21,27,33,39,45,51,57,63,69,75,81,87,93,99,105,
111,117,123,129,135,141}
};


#define ZCODE_LEN 32


void ZCODE_E_Append(uint8_t * src_buf, uint8_t * dst_buf, _Bool parity)
{
    uint8_t b0;
    uint8_t b1;
    uint8_t bprev;
    uint8_t res;


    uint8_t code[4][9] = {{0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0},{0,
0,0,0,0,0,0,0,0}};


    for(int j=0; j<4; j++)
    {
        for(uint8_t i=0; i<72; i++)
        {
            b0      = (src_buf[ n_e[j][ i ]  /8] << ( n_e[j][ i ]  %8)) & 0x80;
            b1      = (src_buf[ n_e[j][72+i] /8] << ( n_e[j][72+i] %8)) & 0x80;
```

```
            bprev   = (code[j][   (i-1)    /8] << (   (i-1)    %8)) & 0x80;
            res     =  b0 ^ b1 ^ bprev;
            code[j][i/8] |= res >> (i%8);
        }
    }


    for(int i=0; i<9;i++)
    {
        dst_buf[i]      = (code[0][i] & (parity ? 0xAA : 0x55)) | (code[1][i] & (parity ?
0x55 : 0xAA));
        dst_buf[i+8]    = (code[2][i] & (parity ? 0xAA : 0x55)) | (code[3][i] & (parity ?
0x55 : 0xAA));
    }
}
```

Приложение Г
(справочное)

Описание транспортных функций и сценариев взаимодействия

Г.1 Обеспечение надежной доставки данных

Для обеспечения надежной доставки данных применяется отправка пакетов в режиме HANDSHAKE_SIMPLE. Отправка выполняется посредством сеансов обмена данными между передающим и приемным узлами. При отправке пользовательских пакетов (длина данных равна параметру MAXLEN) количество пакетов, входящих в сеанс отправки, определяется параметром MACK. Значения MACK более единицы позволяют уменьшить количество отправляемых пакетов подтверждения приема. При групповой отправке количество пакетов, входящих в сеанс отправки, равно количеству пакетов в группе. Каждый отправляемый пакет содержит в своем заголовке HEADER поле ITER, которое инкрементируется для каждого последующего пакета. Значения поля ITER изменяются в диапазоне 0—31. Передающий узел (устройство или сервер), выполняя отправку последнего пакета из группы, выставляет в нем флаг ACK в заголовке пакета. Данный флаг сигнализирует приемной стороне, что необходимо отправить в ответ системный пакет ACK_P.

ACK_P содержит 32-битную маску, каждый бит которой содержит информацию об успешном приеме пакета с номером итератора, соответствующим позиции бита в маске.

Передающий узел обрабатывает ACK_P-пакет и повторно отправляет один или несколько пакетов, которые не были доставлены и которые соответствуют данному сеансу отправки.

Если передающий узел не получает ACK_P-пакет в ответ на запрос, он повторяет отправку последнего пакета по истечении заданного интервала времени (тайм-аута) NBFI_RX_TIMEOUT, вычисляемого по формуле

$$NBFI_RX_TIMEOUT = NBFI_UL_DELAY + NBFI_DL_LISTEN_TIME + random()\%NBFI_DL_ADD_RND_LISTEN_TIME \quad (Г.1)$$

Параметры NBFI_UL_DELAY, NBFI_DL_LISTEN_TIME и NBFI_DL_ADD_RND_LISTEN_TIME зависят от выбранных режимов скорости передачи данных. Их значения приведены в таблицах Г.1—Г.3.

Таблица Г.1 — Значение параметра NBFI_UL_DELAY в зависимости от режима скорости передачи данных

NBFI_TX_PHY_CHANNEL	TIMEOUT, с
UL_DBPSK_50_PROT_E	30
UL_PSK_200	30
UL_DBPSK_400_PROT_E	5
UL_PSK_500	5
UL_DBPSK_3200_PROT_E	1
UL_PSK_5000	1
UL_DBPSK_25600_PROT_E	0,5
UL_PSK_FASTDL	0,5

Таблица Г.2 — Значение параметра NBFI_DL_LISTEN_TIME в зависимости от режима скорости передачи данных

NBFI_RX_PHY_CHANNEL	TIMEOUT, с
DL_PSK_200	40
DL_PSK_500	40
DL_PSK_5000	40
DL_PSK_FASTDL	40

Таблица Г.3 — Значение параметра NBFI_DL_ADD_RND_LISTEN_TIME в зависимости от режима скорости передачи данных

NBFI_RX_PHY_CHANNEL	TIMEOUT, с
DL_PSK_200	20
DL_PSK_500	20
DL_PSK_5000	20
DL_PSK_FASTDL	20

Повторные отправки пакетов выполняются до тех пор, пока не будет получен ответ либо не будет превышено число повторных отправок, заданное параметром NUM_OF_RETRIES. Повторная отправка пакетов, вызванная получением ACK_P-пакета, также входит в расчет общего числа повторов и ограничена параметром NUM_OF_RETRIES.

В случае успешной отправки всех пакетов, входящих в сеанс, передающий узел отправляет системный пакет CLEAR, информирующий приемный узел, что все данные переданы и можно выполнить очистку истории принятых данных.

CLEAR-пакет не является строго обязательным для каждого сеанса отправки, по сути, для успешной работы протокола получателю данных достаточно одного CLEAR-пакета на один полный оборот итератора входных пакетов (32 пакета). Поэтому для снижения объема отправляемых данных можно не выполнять отправку CLEAR-пакетов для одиночных пакетов либо небольших групп пакетов.

При неуспешном выполнении сеанса отправки данных от устройства к серверу передающий узел выполняет сброс всех параметров работы NB-Fi к значениям по умолчанию и отсылает системный пакет CONF с полем данных NBFI_PARAM_MODE, уведомляя сервер о смене режима работы. Сброс параметров и отправка CONF-пакета может быть отключена при помощи флага FLG_NO_RESET_TO_DEFAULTS, содержащегося в параметре NBFI_ADDITIONAL_FLAGS.

На рисунке Г.1 приведен пример успешной отправки группового пакета*. В рамках данного сеанса выполнена отправка пяти пакетов с итераторами 12—16. В ответ на пакет с итератором 16, содержащий флаг ACK, сервер отправил ACK_P-пакет с маской, сообщающей, что сообщения с итераторами 12—16 успешно приняты. Приняв данный пакет, передающий узел отправил CLEAR-пакет, означающий успешное завершение сеанса.

```
(707A99_UL) UL S-M - 12 - 05243359D3C28755 - New group. mark all old messages assembled - UL_DBPSK_400_PROT_D
(707A99_UL) UL --M - 13 - 00F259D3C28A5000 - '\x00\xf2Y\xd3\xc2\x8aP\x00' - UL_DBPSK_400_PROT_D
(707A99_UL) UL --M - 14 - 0059D3C28A540000 - '\x00Y\xd3\xc2\x8aT\x00\x00' - UL_DBPSK_400_PROT_D
(707A99_UL) UL --M - 15 - 59D3C2945100F359 - 'Y\xd3\xc2\x94Q\x00\xf3Y' - UL_DBPSK_400_PROT_D
(707A99_UL) UL -AM - 16 - D3C2945500F3D3A5 - 3359D3C2875500F259D3C28A50000059D3C28A54000059D3C2945100F359D3C2945500F3
(707A99_DL) DL S-- - 16 - 000000000F110040 - 9197:0 - Acked [12, 13, 14, 15, 16], SNR 17,  - DL_PSK_FASTDL
(707A99_UL) UL S-- - 16 - 0400000000000000 - OK, server clears UL history - UL_DBPSK_400_PROT_D
```

Рисунок Г.1 — Пример успешной отправки группового пакета

На рисунке Г.2 приведен пример успешной отправки группового пакета с повторной отправкой пакетов. В рамках данного сеанса выполнена повторная отправка пакета с итератором 13, так как в ответ на него не был получен ACK_P-пакет (несмотря на то что он был отправлен). Также выполнена повторная отправка пакета с итератором 11, так как данный пакет не принят сервером и не перечислен в маске ACK_P-пакета.

```
(70B4F6_UL) UL S-M - 10 - 051D32038259D27E - New group. mark all old messages assembled - UL_DBPSK_50_PROT_D
(70B4F6_UL) UL --M - 12 - 0000000286000000 - '\x00\x00\x00\x02\x86\x00\x00\x00' - UL_DBPSK_50_PROT_D
(70B4F6_UL) UL -AM - 13 - 0002880000000000 - '\x00\x02\x88\x00\x00\x00\x00\x00' - UL_DBPSK_50_PROT_D
(70B4F6_DL) F9D47446A8E91A037094
(70B4F6_DL) DL S-- - 13 - 00000000050F00C0 - 9191:0 - Acked [10, 12, 13], SNR 15,  - DL_PSK_200
(70B4F6_UL) UL -AM - 13 - 0002880000000000 - Repeat, DL 26dBm,  - UL_DBPSK_50_PROT_D
(70B4F6_DL) F9D47446A8EB1A03D154
(70B4F6_DL) DL S-- - 13 - 00000000050D00C0 - 9191:0 - Acked [10, 12, 13], SNR 13,  - DL_PSK_200
(70B4F6_UL) UL -AM - 11 - A000098BBF028400 - 32038259D27EA000098BBF028400000000028600000000028800000000 [10, 11, 12, 13]
(70B4F6_DL) 69ACD8132544F718C060
(70B4F6_DL) DL S-- - 11 - 00400000010C00C0 - 9191:0 - Acked [10, 11, 12], SNR 12,  - DL_PSK_200
(70B4F6_UL) UL S-- - 11 - 0400000000000000 - OK, server clears UL history - UL_DBPSK_50_PROT_D
```

Рисунок Г.2 — Пример успешной отправки группового пакета с повторной отправкой пакетов

* *Взят из log-файла обмена на телеком-сервере компании ООО «Телематические Решения» (бренд «WAVIoT»).*

В режиме работы HANDSHAKE_NONE передающий узел не выполняет запрос ACK_P-пакетов, и передача данных в этом случае выполняется без контроля успешной доставки (верификации).

Г.2 Передача групповых пакетов

Пакеты данных большой длины (не умещающиеся в одном пакете MAC-уровня) могут быть отправлены при помощи отправки группы пакетов. Максимальное число пакетов в группе равно 30. Для UPLINK-пакетов с параметром MAXLEN, равным 8, суммарная максимальная длина пакета составляет 240 байт.

Групповая отправка может быть выполнена как в режимах работы с верификацией доставки данных, так и без нее. Первый пакет в группе является системным пакетом типа GROUP и содержит информацию об общей длине группы, а также контрольную сумму данных, содержащихся в группе. Размер контрольной суммы составляет 1 байт. Каждый пакет группы содержит флаг MULTI в своем заголовке. На приемной стороне выполняются обработка входных пакетов и «склеивание» пакетов в один блок данных. Программный код реализации функции вычисления контрольной суммы на языке Си приведен в разделе Б.2 приложения Б.

Г.3 Передача коротких пакетов (длиной менее MAXLEN)

Пакеты, имеющие длину менее чем MAXLEN, отправляются при помощи системных SHORT-пакетов. Длина полезных данных указана в младших 7 битах первого байта данных SHORT-пакета. Данный параметр может иметь значения от 1 до 127, но не более чем MAXLEN-1.

Г.4 Разрешение коллизий одновременной передачи

Базовые станции стандарта NB-Fi позволяют выполнять одновременный прием и передачу множества каналов. Это позволяет не учитывать при реализации транспортного уровня коллизии, вызванные одновременной передачей пакетов различными устройствами. Также прием данных базовой станцией выполняется без снижения качества в момент передачи данных на устройства.

В противовес базовым станциям конечные устройства NB-Fi работают в режиме «полудуплекс», выполняя единовременно либо прием, либо передачу данных.

Коллизии возникают в тот момент, когда базовая станция и устройство выполняют встречную передачу пакетов.

Для минимизации вероятности возникновения данных коллизий поведение устройств и сервера должно определяться с учетом следующих правил:

1) При отправке очередного пакета данных передающая сторона всегда выставляет в заголовке пакета флаг MULTI в том случае, если вслед за данным пакетом планируется отправка следующего. Приемная сторона, принимая пакет с флагом MULTI, должна продолжать выполнять прием и не выполнять передачу до получения пакета без флага MULTI либо до истечения заданного интервала времени (тайм-аута) NBFI_DL_LISTEN_TIME. Величина данного тайм-аута различается для разных скоростей приема.

2) При повторных отправках пакетов по причине неполучения ACK_P-пакета в ответ на запрос флага ACK используемый тайм-аут должен иметь различные значения от повтора к повтору. Это позволяет избежать циклического повторения коллизии из-за постоянства значений тайм-аутов. Переменная составляющая тайм-аута повторов определяется параметром NBFI_DL_ADD_RND_LISTEN_TIME. Величина данного тайм-аута различается для разных скоростей приема.

Г.5 Особенности работы в режиме DRX

Данный режим предназначен для устройств, которые должны обладать очень малыми значениями среднего потребления энергии (например, устройства с батарейным питанием). Проблема реализации радиосвязи в таких устройствах заключается в невозможности выполнения приема данных в непрерывном режиме. DRX — это режим, который специально разработан с целью решить данную проблему максимально эффективно.

Отличие от режима CRX состоит в том, что устройство кратковременно переходит в режим приема сразу после окончания передачи последнего пакета из группы пакетов, отправляемых непрерывно друг за другом. Сервер выполняет отправку данных только во время действия данного «временного окна».

Таким образом, сеанс обмена данными происходит по инициативе устройства. Сервер может инициировать дополнительные сеансы обмена, используя флаг MULTI в заголовке последнего пакета, отправляемого устройству в рамках предыдущего сеанса.

Сеанс отправки данных от устройства к серверу выполняется в полном объеме до подтверждения приема всех пакетов. Следующий за ним сеанс отправки данных от сервера к устройству может быть прерван из-за потерь данных при плохом уровне сигнала и затем продолжен при следующем «выходе устройства на связь».

Г.6 Синхронизация системного времени

При обмене данными в режиме подтверждения доставки возможна реализация механизма синхронизации времени между сервером и конечными устройствами. Сценарий взаимодействия между сервером и устройством следующий:

- конечное устройство, выполнив отправку данных и получив пакеты, подтверждающие их доставку, отправляет системный пакет CLEAR_T, информирующий приемный узел (сервер) о том, что все данные переданы и

можно выполнить очистку истории принятых данных. Помимо этого, в данном пакете содержится значение текущего (на момент отправки пакета) времени системных часов передающего узла (устройства). Время передается в виде 4-байтного значения, соответствующего формату времени Unix Timestamp (количество секунд от 1 января 1970 года) и часовому поясу UTC.

Сервер сравнивает полученное время с собственным системным временем и в том случае, если разница превышает значение 8191 с, отсылает системный пакет SENDTIME, содержащий значение текущего времени в формате Unix Timestamp. Пакет отсылается без запроса подтверждения.

Если разница не превышает значение 8191 с, но более 30 с, то сервер запоминает это значение для того, чтобы при следующем сеансе обмена выполнить коррекцию системного времени устройства, отправив величину данной поправки внутри пакета ACK_P.

Таким образом, при регулярной отправке данных в режимах работы CRX и DRX выполняется синхронизация времени конечного устройства с точностью (с учетом возможных задержек при прохождении данных через очередь отправки базовой станции и сеть Интернет) от нескольких секунд до нескольких минут. Преимуществом данного механизма является практически полное отсутствие отправки дополнительных пакетов в обоих направлениях.

Г.7 Смена ключа шифрования MAC-уровня

Выделение нового ключа шифрования выполняет конечное устройство по собственной инициативе либо по запросу от сервера при помощи системного пакета GETNEWKEY. Устройство генерирует случайное число размером 32 байта и отправляет его на сервер в виде пяти системных пакетов KEY0—KEY4. При этом шифрование передаваемой информации обеспечивается с использованием старого ключа шифрования.

Сервер, получив ключ шифрования, выполняет отправку пакета CONF (код параметра NBFI_APPLY_KEY), содержащего контрольную сумму ключа размером 4 байта. Устройство, получив этот пакет, сверяет контрольную сумму и в случае совпадения применяет новый ключ. Данный CONF-пакет отправляется сервером командой WRITE_CMD_ACK, что обеспечивает двойное квитирование доставки.

Г.8 Описание способов идентификации устройств

Для идентификации типа устройства необходимо использовать три строковых идентификатора, длина которых не должна превышать 128 байт:

- идентификатор производителя — параметр, уникальный для каждого производителя устройств;
- идентификатор устройства — параметр, уникальный для каждого типа устройств;
- идентификатор протокола — параметр, однозначно определяющий формат данных и логику работы устройства на уровне приложения.

Данные идентификаторы сервер приложений запрашивает у устройства при их отсутствии в базе данных сервера либо устройство отправляет их самостоятельно при первом включении периодически или при изменении значений.

Для получения значений идентификаторов сервер должен отправить пакет уровня представления с командами NBFI_PL_GET_MAN_ID, NBFI_PL_GET_HW_ID и NBFI_PL_GET_PROT_ID. Устройство должно прислать ответный пакет с таким же кодом команды и со значением соответствующего идентификатора в поле DATA.

Сервер приложений использует данные идентификаторы для определения, поддерживается ли данное устройство и каким образом необходимо выполнять взаимодействие с данным устройством.

Г.9 Описание механизма изменения типа шифрования и смены ключа на уровне представления

Возможно использование трех режимов работы шифрования на уровне представления: без шифрования, «Магма», а также режим шифрования с использованием другого выбранного алгоритма (два варианта выбора). Тип шифрования, который использован для шифрования поля DATA, всегда должен быть указан в поле ENC.

Для систем, к которым предъявляются требования в части безопасности передачи данных, включенный режим шифрования на уровне представления является обязательным, поскольку он обеспечивает достоверную идентификацию устройства и целостность данных.

Для смены типа используемого шифрования либо его включения или отключения сервер приложения должен отправить одну из команд:

NBFI_PL_ENC_SET_NONE, NBFI_PL_ENC_SET_MAGMA, NBFI_PL_ENC_SET_CUSTOMCRYPTO1, NBFI_PL_ENC_SET_CUSTOMCRYPTO2.

Если устройство не поддерживает данный тип шифрования, то оно не меняет режим шифрования, и данные продолжают поступать с прежним значением поля ENC.

Выделение нового ключа шифрования выполняет конечное устройство по запросу от сервера при помощи пакета с кодом команды NBFI_PL_ENC_NEW_KEY. Устройство генерирует случайное число размером 32 байта и отправляет его на сервер внутри поля DATA.

Сервер, получив ключ шифрования, выполняет отправку пакета с кодом команды NBFI_PL_ENC_APPLY_KEY, содержащего в поле DATA контрольную сумму ключа размером 4 байта. Устройство, получив этот пакет, сверяет контрольную сумму и в случае совпадения применяет новый ключ.

Приложение Д
(справочное)

Описание механизма автоматического выбора оптимальной скорости передачи данных

Устройства, работающие в режимах DRX и CRX, выполняют автоматический контроль качества радиосигнала (как приема, так и передачи) и производят смену скоростей передачи данных, опираясь на средние значения соотношений сигнал/шум (SNR) для UPLINK-пакетов и DOWNLINK-пакетов.

Расчет SNR на входе приемника устройства выполняется при приеме пакетов, используя данные, предоставляемые аппаратурой радиотрансивера, входящего в состав устройства. Расчет SNR выходного сигнала выполняется на основании данных, которые измеряет базовая станция на входе своего приемника и которые передаются устройству в одном из полей ACK_P-пакета.

Решения о смене скорости передачи данных UPLINK-пакетов либо DOWNLINK-пакетов принимает устройство, анализируя уровни SNR.

При достаточно высоком значении SNR (≥ SNRLEVEL_FOR_UP + TXSNRDEG либо ≥ SNRLEVEL_FOR_UP + RXSNRDEG) выполняется перевод скорости на один уровень выше. Так как сервер принимает одновременно все поддерживаемые скорости передачи UPLINK-пакетов, смена скоростей UPLINK осуществляется без уведомления сервера. При смене скорости DOWNLINK-пакетов выполняется отправка системного CONF-пакета с полем данных NBFI_PARAM_MODE, содержащим новые параметры скоростного режима. Данный пакет формируется с флагом ACK и требует подтверждения приема. Прием пакета ACK_P устройство выполняет уже на новой скорости. В отсутствие ответного пакета выполняются повторные отправки CONF-пакетов. Если после определенного числа повторных попыток доставки CONF-пакета, равного NUM_OF_RETRIES, пакет ACK_P так и не был получен, устройство возвращает свой скоростной режим в предыдущее состояние, отсылая CONF-пакет с полем данных NBFI_PARAM_MODE, содержащим предыдущие значения параметров скорости. В этот раз CONF-пакет отсылается единожды без запроса подтверждения.

Повышение скорости UPLINK либо DOWNLINK выполняется при условии, что базовая станция поддерживает более высокие скоростные режимы. Об этом сервер уведомляет устройство, выставляя флаги UL_SPEED_NOT_MAX и DL_SPEED_NOT_MAX, содержащиеся в пакете ACK_P.

Устройство повышает скорость передачи данных до тех пор, пока уровень SNR не снизится до значения, не позволяющего выполнять дальнейшее повышение, либо пока не будет достигнуто максимальное значение скорости.

Если при достижении максимального уровня скорости передачи UPLINK-пакетов уровень SNR имеет достаточное значение (≥ SNRLEVEL_FOR_UP + TXSNRDEG), устройство выполняет постепенное (с шагом 3 дБ) снижение мощности передатчика.

Если при достижении максимального уровня скорости приема DOWNLINK-пакетов уровень SNR имеет достаточное значение (≥ SNRLEVEL_FOR_UP + RXSNRDEG), устройство уведомляет сервер о необходимости снижения выходной мощности передатчика базовой станции, выставляя флаг DL_POWER_STEP_DOWN в пакете ACK_P, отправляемом от устройства к серверу.

Если уровень SNR снижается ниже заданного значения (SNRLEVEL_FOR_DOWN), устройство выполняет обратные действия. Сначала повышается выходная мощность передатчиков устройства или базовой станции, затем выполняется ступенчатое снижение скорости передачи либо приема данных.

Для уведомления сервера о необходимости повышения выходной мощности используется флаг DL_POWER_STEP_UP в пакете ACK_P, отправляемом от устройства к серверу.

Пороговые уровни для повышения и понижения скоростей либо мощности и поправочные коэффициенты для различных скоростей приведены в таблицах Д.1—Д.3.

Таблица Д.1 — Пороговые уровни для повышения и понижения скоростей либо мощности

Параметр	Значение порога, дБ
SNRLEVEL_FOR_UP	15
SNRLEVEL_FOR_DOWN	10

Таблица Д.2 — Значение поправочного коэффициента TXSNRDEG для различных скоростей передачи

NBFI_TX_PHY_CHANNEL	Поправочное значение, дБ
UL_DBPSK_50_PROT_E	0

Окончание таблицы Д.2

NBFI_TX_PHY_CHANNEL	Поправочное значение, дБ
UL_DBPSK_400_PROT_E	9
UL_DBPSK_3200_PROT_E	18
UL_DBPSK_25600_PROT_E	27

Таблица Д.3 — Значение поправочного коэффициента RXSNRDEG для различных скоростей приема

NBFI_RX_PHY_CHANNEL	Поправочное значение, дБ
DL_PSK_200	0
DL_PSK_500	4
DL_PSK_5000	14
DL_PSK_FASTDL	25

Механизм автоматического выбора скоростного режима работы устройства может быть отключен при помощи флага FLG_FIXED_BAUD_RATE, содержащегося в параметре NBFI_ADDITIONAL_FLAGS.

Механизм автоматического управления мощностью передачи может быть отключен при помощи флага FLG_NO_REDUCE_TX_PWR, содержащегося в параметре NBFI_ADDITIONAL_FLAGS.

На рисунке Д.1 приведен пример сеанса обмена пакетами, выполняющий повышение скоростей передачи и приема*.

```
(6FB2EC_UL) UL SA- - 01 - 0100A21C00000E0F - Modem info: 3.34V, 28°C, Average SNR DL 0, UL 0, Noise -136, UL 15dBm
(6FB2EC_DL) 3BDE74B39C62C70B518E
(6FB2EC_DL) DL S-- - 01 - 00000000004500C0 - 9297:0 - Acked [], SNR 69,  - DL_PSK_200
(6FB2EC_UL) UL SA- - 02 - 0680010118010F02 - PHY mode: UL_DBPSK_400_PROT_D & DL_PSK_500 26dBm - UL_DBPSK_400_PROT_D
(6FB2EC_DL) 007742BBD40A4534A02E
(6FB2EC_DL) DL S-- - 02 - 0000000000460040 - 9297:0 - Acked [], SNR 70,  - DL_PSK_500
(6FB2EC_UL) UL SA- - 02 - 0680010118020F02 - PHY mode: UL_DBPSK_400_PROT_D & DL_PSK_5000 26dBm - UL_DBPSK_400_PROT_D
(6FB2EC_DL) 007742BBD47745343036
(6FB2EC_DL) DL S-- - 02 - 00000000003B0040 - 9297:0 - Acked [], SNR 59,  - DL_PSK_5000
(6FB2EC_UL) UL SA- - 02 - 0680010118030F02 - PHY mode: UL_DBPSK_400_PROT_D & DL_PSK_FASTDL 26dBm - UL_DBPSK_400_PROT_D
(6FB2EC_DL) 007742BBD47245342037
(6FB2EC_DL) DL S-- - 02 - 00000000003E0040 - 9297:0 - Acked [], SNR 62,  - DL_PSK_FASTDL
```

Рисунок Д.1 — Пример сеанса обмена пакетами, выполняющий повышение скоростей передачи и приема

* *Взят из log-файла обмена на телеком-сервере компании ООО «Телематические Решения», (бренд «WAVIoT»).*

Приложение Е
(справочное)

Настройка параметров NB-Fi

Режимы функционирование протокола NB-Fi конкретного устройства определены совокупностью параметров, объединенных в единую структуру данных NBFI_SETTINGS. Некоторая часть данных параметров должна быть известна серверу для обеспечения нормальной работы протокола. В частности, такими параметрами являются следующие: режим работы устройства, текущие скорости передачи данных, перечень поддерживаемых устройством скоростей, рабочий диапазон частот и т. д. Данные параметры записываются в базу данных сервера при выделении лицензии на конкретные устройства и затем в процессе работы синхронизируются с соответствующими значениями этих параметров в устройстве посредством отправки системных пакетов.

Все параметры NB-Fi устройства доступны для чтения и записи при помощи системных CONF-пакетов. Формат данных пакетов и наименование параметров описаны в 7.2.

Приложение Ж
(справочное)

Защита данных в протоколе NB-Fi

Шифрование данных выполняется на двух уровнях — MAC-уровне и на уровне представления.

Шифрование на MAC-уровне предназначено для защиты механизма обмена данными на транспортном уровне.

Шифрование данных выполняется как для UPLINK-пакетов, так и для DOWNLINK-пакетов.

Для шифрования данных на MAC-уровне применяется алгоритм «Магма», используются уникальные ключи длиной 256 бит.

Пакеты данных UPLINK и DOWNLINK содержат поле, соответствующее контрольной сумме незашифрованных данных. Это позволяет, выполнив дешифрование пакета, проверить валидность полученных данных и отбросить пакет в случае несовпадения ключей сервера и модема. Данный механизм делает систему связи устойчивой не только к попыткам несанкционированного получения данных, но и к попыткам злонамеренного заполнения эфира пакетами со случайным содержанием в поле Payload с целью нарушения нормальной работы устройств.

Возможны два сценария присвоения ключей устройствам — «прошивка» ключей при производстве либо выделение новых ключей при подключении устройства к сети.

В первом варианте ключи шифрования генерируются случайным образом при выделении лицензии на сервере оператора сети. Каждому ID (идентификатору) NB-Fi сопоставляется персональный ключ, который сохраняется на сервере и в выходном файле лицензии, используемом впоследствии при «прошивке» программ в модемы устройств. Данный вариант обеспечивает контроль над уникальностью идентификаторов NB-Fi, выполняет защиту от несанкционированной передачи под «чужим» ID на любом этапе взаимодействия устройства с сервером. Возможен также сценарий «прошивки» в устройство временного ключа шифрования, который при подключении устройства к сети будет изменен по инициативе сервера или устройства.

Во втором варианте устройства изначально не содержат ключ шифрования, и при подключении к сети используется «открытый» обмен данными. По инициативе устройства выполняется регистрация нового идентификатора на сервере и выделяется ключ шифрования. Данная процедура выполняется успешно только в случае отсутствия на сервере информации об уже зарегистрированном устройстве с таким идентификатором.

Шифрование данных на уровне представления позволяет выполнять дополнительное шифрование пользовательских данных с целью сделать их недоступными для оператора сети, обслуживающего транспортный сервер, но не владеющего сервером приложений для данного устройства.

Выделение ключей шифрования выполняется сервером приложений по инициативе сервера.

Используемые алгоритмы шифрования: «Магма», другой выбранный алгоритм (см. 8.1, 8.2.3).

Размер ключа шифрования — 256 бит.

Помимо шифрования на уровне представления выполняется присвоение каждому следующему пакету данных нового идентификатора (итератора) размером 16 бит, что позволяет реализовать защиту данных от атаки повторного воспроизведения.

Приложение И
(справочное)

Примеры использования протокола NB-Fi

В настоящем приложении приведен пример создания решения по удаленному сбору данных на протоколе NB-Fi для жилищно-коммунального хозяйства (ЖКХ) и городского хозяйства [2].

И.1 Описание системы

LPWAN-сеть, использующая протокол связи NB-Fi в нелицензируемом диапазоне частот, разработана российской компанией и успешно применяется в России и за рубежом, с суммарным числом установленных устройств в 2017 году более 100 тыс. шт. Компания реализовала проекты по созданию удаленного сбора данных с приборов учета ресурсов ЖКХ в более чем 50 городах России.

Высокая дальность связи и большая емкость сети позволяют обеспечивать сбор данных с большого количества устройств при помощи одной базовой станции.

Например, типовой случай использования данной технологии — установка базовой станции на крыше одного из зданий микрорайона и счетчиков воды и электроэнергии в каждую квартиру близлежащих домов в радиусе нескольких километров. При этом осуществляется 100%-ный сбор данных со всех устройств, включая данные с устройств, установленных в металлические щиты и полуподвальные помещения.

И.2 Архитектура системы

Архитектура системы состоит из четырех уровней и включает в себя:

1) устройства, отправляющие и принимающие данные по протоколу стандарта NB-Fi;

2) базовые станции, осуществляющие прием, обработку и передачу сообщений от устройств (UPLINK-пакеты) на телеком-сервер, а также отправку нисходящих сообщений на устройства (DOWNLINK-пакеты);

3) телеком-сервер, осуществляющий прием, обработку и хранение сообщений от всех базовых станций для всех устройств, а также обеспечивающий интеграцию исходных данных с сервером приложений и со сторонними программно-техническими комплексами;

4) сервер приложений, осуществляющий отображение данных от устройств клиентам и предоставляющий возможность выгрузки отчетов в требуемом виде.

И.3 Описание устройств

Компания самостоятельно производит и разрабатывает все компоненты системы: устройства, базовые станции, серверное программное обеспечение. В настоящий момент компания использует микроконтроллер STM32 с радиотрансивером компании ON Semiconductor AX5043, разработав для данной аппаратной платформы ПО с реализацией протокола передачи данных по стандарту NB-Fi.

Компания производит устройства, отправляющие и принимающие данные по протоколу стандарта NB-Fi, среди которых:

- счетчики воды с накладным внешним модемом;
- счетчики воды со встроенным в основную плату модемом;
- однофазные и трехфазные счетчики электрической энергии со встроенным модемом;
- теплосчетчики со встроенным и внешним модемом;
- счетчики газа со встроенным модемом;
- радиомодемы, подсоединяющиеся к импульсным или цифровым выводам внешних устройств.

Перечисленные выше устройства выпускаются с автономным питанием (от батареи), при этом существует возможность использовать и внешнее питание.

Счетчики воды, газа, тепла отправляют сообщения два раза в сутки и содержат внутри себя данные о почасовом потреблении энергоресурсов.

Счетчики воды с накладным модемом получают показания о потреблении при помощи оптического датчика, который считывает вращение бегунка. Накопленное число вращений передается на сервер приложений, где складывается с начальными показаниями данных счетчика.

Аналогично работает радиомодем, считывающий импульсы с импульсных выходов устройств: накопленные значения импульсов передаются на сервер приложений, где суммируются с начальными показаниями.

Счетчики воды, тепла, газа со встроенным радиомодулем два раза в сутки передают актуальные показания с почасовой разбивкой. Счетчики со встроенным модемом имеют жидкокристаллический дисплей, отображающий показания, и именно эти показания и передаются на сервер приложений.

Счетчики тепла, электроэнергии с внешним радиомодемом с цифровым (RS-485) выходом считывают показания с прибора учета по цифровому интерфейсу. Для отдельных приборов учета разработаны индивидуальные радиомодемы, позволяющие осуществить монтаж радиомодема под крышку устройства. Радиомодемы с батарейным питанием в зависимости от конфигурации отправляют данные с периодичностью от одного раза в месяц до

двух раз в сутки. Существует модель радиомодема для электросчетчика с питанием от сети, что позволяет увеличить срок службы радиомодема.

Указанные выше устройства отправляют на сервер только восходящие (UPLINK) сигналы без подтверждения доставки. В случае пропуска сигнала в определенный период на сервере приложений данных за этот период не окажется, и при следующем получении сигнала на сервере приложений будут отображены только имеющиеся данные (последнее значение потребления на приборе учета и почасовая разбивка потребления за периоды, по которым данные переданы).

Выпускаемые компанией электросчетчики осуществляют отправку сообщений на сервер один раз в час. Сообщения содержат данные о потреблении электроэнергии по каждой фазе, а также прочие параметры сети. Устройства работают в режиме CRX (Continuous RX) и обеспечивают постоянный прием нисходящих (DOWNLINK) сообщений от сервера. Отправка данных на электросчетчик с сервера возможна в любой момент. Данный режим работы позволяет отправлять подтверждение доставки сигнала, а также конфигурировать электросчетчик и осуществлять управление реле электросчетчика.

Мощность излучения устройств, как правило, составляет 15 мВт при максимально разрешенном уровне излучения 25 мВт. Расчетный срок работы устройств от батареи составляет свыше 10 лет. Рабочий температурный диапазон устройств составляет от минус 40 °С до плюс 70 °С.

Примеры реализации конечных устройств (импульсного модема и платы комплекта разработчика), включающие проекты печатных плат и программное обеспечение в исходных кодах, а также библиотеку реализации протокола NB-Fi, содержатся в файловом архиве [3].

И.4 Описание базовых станций

Базовая радиостанция представляет собой стационарный приемопередатчик радиосигнала. Для приема восходящих (UPLINK) пакетов данных со стороны базовой станции применяется принцип SDR-систем (software-defined radio), где входной радиосигнал оцифровывается во всей полосе приема 50 кГц и в дальнейшем подвергается программной обработке. Прием пакетов выполняется базовой станцией по всей полосе частот, при этом выделяются одновременно пакеты, имеющие различные скорости. Теоретическое количество каналов составляет 1024 для скорости 50 бит/с, и 128, 16 и 2 для скоростей 400, 3200 и 25600 бит/с соответственно.

Передача нисходящих (DOWNLINK) пакетов выполняется при помощи цифрового модулятора сигнала, позволяющего передавать одновременно несколько узкополосных каналов, если суммарная мощность передачи не превышает установленного значения. Передача осуществляется в полосе 100 кГц.

За счет разнесения полос приема и передачи по частоте и развязки приемной и передающих антенн по поляризации достигается одновременный прием и передача (полный дуплекс) без ухудшения характеристик радиосвязи.

Габаритные размеры корпуса базовой станции (высота × ширина × глубина, мм) составляют 200 × 120 × 75. Рабочие условия эксплуатации базовой радиостанции позволяют устанавливать базовую станцию на улице — температура окружающей среды от минус 50 °С до плюс 70 °С, относительная влажность (без конденсации влаги) от 40 до 98 %. Степень защиты корпуса от проникновения пыли и воды соответствует IP67 по ГОСТ 14254.

Базовая станция имеет вычислительное ядро FPGA Artix 100. Характеристики производительности базовой станции:

- число арифметических операций — свыше 7500 MFLOPS (млн операций/с);
- скорость произвольного доступа к памяти — свыше 100 Гбит/с;
- производительность декодера помехоустойчивого кода — не более 40 000 гипотез/с;
- суммарная информационная скорость декодера помехоустойчивого кода — свыше 5 Мбит/с;
- чувствительность базовой станции — минус 150 dBm (для скорости 50 бит/с).

В зависимости от комплекта поставки базовая станция может комплектоваться одной или несколькими антеннами: принимающая коллинеарная антенна (для приема UPLINK-пакетов), антенна, передающая петлевой вибратор (для отправки DOWNLINK-пакетов), двухдиапазонная секторная антенна (для UPLINK- и DOWNLINK-пакетов).

В зависимости от исполнения телеком-сервер и сервер приложения могут быть реализованы прямо на базовой станции, что обуславливается требованиями заказчиков к автономности и изолированности используемых систем.

Пример реализации базовой станции, построенной на базе простого SDR-приемника и одноплатного встраиваемого микрокомпьютера Raspberry PI, содержится в файловом архиве [3].

В данном решении прием UPLINK-пакетов выполняется при помощи программной обработки записанного сигнала с поддержкой скоростей приема 50 и 400 бит/с. Передача DOWNLINK-пакетов выполняется при помощи радиотрансивера компании ON Semiconductor AX8052F143 с поддержкой всех описанных в настоящем стандарте скоростей передачи.

В файловом архиве содержатся проект печатной платы SDR-приемника и программное обеспечение базовой станции в исходных кодах, написанное на языке программирования C++ с использованием библиотеки Qt5.

И.5 Описание телеком-сервера

Функциями телеком-сервера являются:

- хранение в базе данных информации об устройствах (идентификаторы, ключи шифрования MAC-уровня, режимы работы транспортного уровня);

- получение восходящих пакетов с данными от множества базовых станций;
- дешифрование данных MAC-уровня и отбрасывание пакетов, не прошедших проверку целостности данных;
- выделение уникальных пакетов (фильтрация и отбрасывание копий одного и того же пакета, принятого разными станциями);
- реализация функций транспортного уровня NB-Fi;
- выбор наилучшей базовой станции для отправки нисходящих пакетов для каждого устройства;
- сохранение пакетов в базу данных;
- взаимодействие с серверным ПО уровня приложений посредством API-интерфейса.

Использование центрального телеком-сервера в качестве ответного узла для всех конечных устройств сети NB-Fi позволяет организовать сеть передачи данных большого радиуса действия, содержащей огромное количество устройств (от сотен тысяч до десятков миллионов). Преимуществом реализации транспортного уровня на стороне сервера является простое разрешение коллизий между пересекающимися по покрытию базовыми станциями при отправке нисходящих пакетов на устройства. В подобной централизованной архитектуре масштабирование сети по расширению покрытия либо по увеличению плотности установки устройств выполняется максимально легко, простым добавлением базовых станций и устройств без необходимости конфигурирования.

Установка телеком-сервера возможна в том числе и на базовые станции. Это позволяет организовать получение/отправку данных непосредственно с базовых станций и на них. При этом возникает необходимость конфигурирования каждой станции в части загрузки в нее информации об устройствах, что делает масштабирование сети проблематичным.

И.6 Описание сервера приложений

Функциями сервера приложений являются:
- поддержка механизма защиты данных уровня представления NB-Fi;
- поддержка конкретных моделей устройств:

а) преобразование пакетов от телеком-сервера в данные для отображения потребителю;

б) преобразование команд от потребителя в пакеты для отправки через телеком-сервер;

в) конфигурирование устройств, обновление ПО, диагностика исправности;

- длительное хранение данных;
- предоставление данных и управление устройствами через API-интерфейс;
- предоставление данных и управление устройствами при помощи графического интерфейса пользователя;
- аналитический и статистический анализ данных, выгрузка отчетов;
- другие функции, зависящие от предметной области данного приложения.

Сервер приложений является верхним уровнем телекоммуникационной системы, построенной на базе сети передачи данных NB-Fi. Данный компонент не является стандартным решением со строго определенным перечнем функций. Его задачи зависят от предметной области, для которой используется система (мониторинг ресурсов ЖКХ, система охранной сигнализации, контроль технологических процессов, мониторинг сельского хозяйства и т. д.). Возможна одновременная работа различных серверов приложения, взаимодействующих с единым телеком-сервером.

Сервер приложений может использовать механизм защиты данных уровня представления, обеспечивая дополнительное шифрование данных приложения, делая их недоступными поставщику телекоммуникационных услуг (владеющему телеком-сервером).

Библиография

[1] gost.e-ecolog.ru
[2] gost.e-ecolog.ru
[3] gost.e-ecolog.ru

УДК 004.738 ОКС 35.020, 35.110

Ключевые слова: информационные технологии, интернет вещей, протокол беспроводной передачи данных, узкополосная модуляция радиосигнала, NB-Fi

БЗ 1—2019/13

Редактор *Л.С. Зимилова*
Технический редактор *И.Е. Черепкова*
Корректор *Е.Р. Ароян*
Компьютерная верстка *Ю.В. Поповой*

Сдано в набор 21.02.2019. Подписано в печать 18.03.2019. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 5,58. Уч.-изд. л. 5,02.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11.
gost.e-ecolog.ru gost.e-ecolog.ru

Создано в единичном исполнении ФГУП «СТАНДАРТИНФОРМ»
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