ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 58603— 2019 (ИСО/МЭК 20922:2016)

Информационные технологии

ИНТЕРНЕТ ВЕЩЕЙ

Протокол организации очередей доставки телеметрических сообщений MQTT. Версия 3.1.1

(ISO/IEC 20922:2016, MOD)

Издание официальное

Москва
Стандартинформ
2019

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Акционерным обществом «Всероссийский научно-исследовательский институт сертификации» (АО «ВНИИС») и Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр» (ООО ИАВЦ) на основе собственного перевода на русский язык англоязычной версии документа, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 022 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 16 октября 2019 г. № 1005-ст

4 Настоящий стандарт является модифицированным по отношению к международному стандарту ИСО/МЭК 20922:2016 «Информационные технологии. Протокол организации очередей доставки телеметрических сообщений MQTT. v3.1.1» (ISO/IEC 20922:2016 «Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1», MOD) путем изменения его структуры для приведения в соответствие с правилами, установленными в ГОСТ 1.5 (подразделы 4.2 и 4.3). Сравнение структуры настоящего стандарта со структурой указанного международного стандарта приведено в дополнительном приложении Д

5 ВВЕДЕН ВПЕРВЫЕ

6 Некоторые положения международного стандарта, указанного в пункте 4, могут являться объектом патентных прав. ИСО и МЭК не несут ответственности за идентификацию подобных патентных прав

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

© ISO, 2016 — Все права сохраняются
© 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 Термины и определения 1
3 Обозначения и сокращения 3
3.1 Обозначения 3
3.2 Сокращения 3
4 Представление данных 3
4.1 Бит 3
4.2 Целочисленные значения данных 3
4.3 Закодированные строки UTF-8 3
5 Формат управляющего пакета MQTT 4
5.1 Структура управляющего пакета MQTT 4
5.2 Фиксированный заголовок 5
5.3 Переменный заголовок 7
5.4 Полезная нагрузка 9
6 Управляющие пакеты MQTT 9
6.1 CONNECT — Клиент запрашивает подключение к Серверу 9
6.2 CONNACK — Подтверждение запроса на подключение 16
6.3 PUBLISH — Публикация сообщения 17
6.4 PUBACK — Подтверждение публикации 20
6.5 PUBREC — Подтверждение получения пакета PUBLISH с QoS 2 21
6.6 PUBREL — Выпуск пакета PUBLISH с QoS 2 21
6.7 PUBCOMP — Публикация завершена 22
6.8 SUBSCRIBE — Подписка на темы 23
6.9 SUBACK — Подтверждение подписки 25
6.10 UNSUBSCRIBE — Отменить подписку на темы 27
6.11 UNSUBACK — Подтверждение отмены подписки 28
6.12 PINGREQ — Запрос PING 29
6.13 PINGRESP — Ответ PING 29
6.14 DISCONNECT — Уведомление об отключении 30
7 Описание работы протоколов 30
7.1 Сохранение состояния 30
7.2 Сетевые подключения 31
7.3 Уровни качества обслуживания и потоки протоколов 31
7.4 Повторная доставка сообщений 34
7.5 Получение сообщения 34
7.6 Порядок отправки сообщений 34
7.7 Названия тем и фильтры тем 35
7.8 Обработка ошибок 37
8 Использование WebSocket в качестве протокола передачи данных 37
8.1 Примечания IANA 37
9 Совместимость 38
9.1 Цели совместимости 38
Приложение А (рекомендуемое) Безопасность 39

Приложение Б (справочное) Обязательные нормативные положения 42
Приложение В (справочное) Предисловие к оригинальному изданию 52
Приложение Г (справочное) Извещение правообладателя оригинального издания 53
Приложение Д (справочное) Сведения об изменениях, внесенных в текст национального стандарта по отношению к исходному стандарту 54
Библиография 55

Введение

MQTT — клиент-серверный протокол передачи сообщений, основанный на модели «издатель — подписчик». Он является легким, открытым, простым, и спроектирован таким образом, чтобы его легко можно было реализовать. Данные характеристики делают его идеальным для использования во многих ситуациях, в том числе в ограниченных средах, например, для организации связи в среде межмашинного взаимодействия (M2M) и Интернета вещей (IoT), где требуется небольшой размер кода и/или пропускная способность сети является приоритетом.

Протокол выполняется с использованием протоколов из набора TCP/IP или через другие сетевые протоколы, которые обеспечивают упорядоченные двунаправленные соединения без потерь. Его функции включают:

а) использование шаблона проектирования «издатель — подписчик», который обеспечивает распределение сообщений по принципу «один ко многим»;

б) доставка сообщений, не зависящая от содержимого информационного наполнения;

в) три уровня качества услуги (QoS) передачи данных:

- «Не более одного раза», когда сообщения доставляются в соответствии с оптимальными затратами операционной среды. Может произойти потеря сообщений. Данный уровень может использоваться, например, для передачи данных датчика температуры окружающей среды, где потеря отдельного измерения не имеет значения, поскольку вскоре после этого будет опубликовано следующее измерение;
- «Не реже одного раза», в этом случае гарантируется доставка сообщений, но возможны дубликаты;
- «Ровно один раз», в этом случае гарантируется доставка сообщений ровно один раз. Данный уровень может использоваться, например, в биллинговых системах, где дублирующиеся или потерянные сообщения могут приводить к применению неправильных сборов;

г) малый объем служебного трафика и протокольные обмены сводятся к минимуму для снижения сетевого трафика;

д) механизм уведомления заинтересованных сторон в случае аварийного разрыва связи.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

ИНТЕРНЕТ ВЕЩЕЙ

Протокол организации очередей доставки телеметрических сообщений MQTT. Версия 3.1.1

Information technology. Internet of things.
Message Queuing Telemetry Transport (MQTT) v3.1.1

Дата введения — 2021—01—01

1 Область применения

Настоящий стандарт определяет требования к разработке Клиента MQTT и реализации Сервера MQTT.

Обязательные нормативные положения настоящего стандарта, составляющие требования к реализации протокола MQTT, отмечены специальными обозначениями, сформированными по шаблону [MQTT-x.x.x-y] (см. раздел 3), и размещенными сразу после соответствующего нормативного положения. Полный перечень обязательных нормативных положений справочно представлен в приложении Б.

Степень значимости отдельных требований и положений настоящего стандарта обозначаются специальными терминами, приведенными в подразделе 2.1 настоящего стандарта.

Критерии оценки соответствия реализации Клиента MQTT или Сервера MQTT требованиям настоящего стандарта, приведеные в разделе 9 настоящего стандарта.

2 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями.

2.1 Ключевые слова «ДОЛЖЕН», «НЕ ДОЛЖЕН», «ТРЕБУЕТСЯ», «БУДЕТ», «НЕ БУДЕТ», «СЛЕДУЕТ», «НЕ СЛЕДУЕТ», «РЕКОМЕНДУЕТСЯ», «МОЖЕТ» и «ОПЦИОНАЛЬНО» в настоящей Спецификации интерпретируются, как описано в IETF RFC (рабочем предложении инженерной рабочей группы Интернета) 2119 [1], следующим образом:

2.1.1 **ДОЛЖЕН** (MUST), а также термины «требуется» (REQUIRED) и «нужно» (SHALL) используется для требований, которые являются абсолютно необходимыми в данной спецификации.

2.1.2 **НЕ ДОЛЖЕН** (MUST NOT) или слова SHALL NOT (не позволяется) означают абсолютный запрет в рамках спецификации.

2.1.3 **СЛЕДУЕТ** (SHOULD), а также глагол рекомендуется (RECOMMENDED) используется для обозначения требований, от выполнения которых можно отказаться при наличии разумных причин. Однако при таком отказе следует помнить о возможных проблемах в результате отказа и принимать взвешенное решение.

2.1.4 **НЕ СЛЕДУЕТ** (SHOULD NOT) и глагол не рекомендуется (NOT RECOMMENDED) используются применительно к особенностям или функциям, которые допустимы и могут быть полезными, но могут вызывать проблемы. При реализации таких опций следует учитывать возможность возникновения проблем и принимать взвешенное решение.

Издание официальное

2.1.5 **МОЖЕТ** (MAY), а также прилагательное необязательный (OPTIONAL) обозначают элементы, реализация которых является необязательной. Одни разработчики могут включать такие опции в свою продукцию для расширения возможностей, а другие — опускать в целях упрощения. Реализация, не включающая ту или иную опцию, должна быть готова к работе с реализациями, которые используют эту опцию (возможно совместная работа будет обеспечиваться за счет некоторого ущерба функциональности). Включающие опцию реализации должны быть готовы (естественно, без использования такой опции) к взаимодействию с реализациями, которые такую опцию не поддерживают.

2.2 **сетевое соединение** (network connection): Структура, предоставляемая базовым транспортным протоколом, который используется MQTT.

- подключает Клиента к Серверу;
- предоставляет средства для отправки упорядоченного потока байтов в двух направлениях без потерь.

Примечание — Примеры приведены в подразделе 7.2.

2.3 **сообщение приложения** (application message): Данные, передаваемые протоколом MQTT по сети для приложения. При передаче сообщений с использованием протокола MQTT обеспечивается соответствующий уровень качества услуг передачи данных и указывается название темы.

2.4 **клиент** (client): Программа или устройство, использующее MQTT. Клиент всегда устанавливает сетевое подключение к Серверу. Клиент может:

- опубликовать сообщения приложений, в которых могут быть заинтересованы другие Клиенты;
- выполнить подписку для запроса сообщений приложения, в получении которых Клиент заинтересован;
- отменить подписку, чтобы удалить запрос на сообщения приложений;
- отключиться от Сервера.

2.5 **сервер** (server): Программа или устройство, которое выступает в качестве посредника между Клиентами, которые публикуют сообщения приложений, и Клиентами, которые выполнили Подписки. Сервер может:

- принимать сетевые подключения от Клиентов;
- принимать сообщения приложений, публикуемые Клиентами;
- обрабатывать запросы на подписку и отмену подписки от Клиентов;
- пересылать сообщения приложений, соответствующие подпискам Клиентов.

2.6 **подписка** (subscription): Подписка включает фильтр по темам и максимальный уровень качества услуг передачи данных. Подписка связана с одним Сеансом. Сеанс может включать несколько Подписок. Каждая Подписка в рамках Сеанса имеет различные фильтры по темам.

2.7 **название темы** (topic name): Метка, прикрепленная к сообщению приложения, сопоставляемая с Подписками, известными Серверу. Сервер отправляет копию сообщения приложения каждому Клиенту, имеющему соответствующую Подписку.

2.8 **фильтр по темам** (topic filter): Выражение, содержащееся в подписке, определяющее интерес к сообщениям одной или нескольких тем. Фильтр темы может содержать подстановочные знаки.

2.9 **сеанс** (session): Межсетевое взаимодействие между Клиентом и Сервером с сохранением состояния. Некоторые Сеансы продолжаются только во время сетевого соединения, другие могут охватывать несколько последовательных сетевых соединений между Клиентом и Сервером.

2.10 **управляющий пакет MQTT** (MQTT control packet): Пакет информации, который отправляется через Сетевое подключение. Спецификация MQTT определяет четырнадцать различных типов управляющих пакетов, один из которых (пакет PUBLISH /ОПУБЛИКОВАТЬ/) используется для передачи сообщений приложений.

2.11 ***последняя воля*** *(will): Механизм протокола MQTT, включающий специальное сообщение приложения и набор характеризующих его параметров, предназначенный для обработки ситуаций аварийного разрыва соединения между Клиентом и Сервером, которые могут возникать при:*

- *ошибках ввода/вывода и отказов сети, обнаруженных Сервером;*
- *превышениях времени ожидания Клиентом;*
- *ошибках соблюдения протокола.*

3 Обозначения и сокращения

3.1 Обозначения

3.1.1 **[MQTT-x.x.x-y]:** Обозначает положения настоящего стандарта, обязательные с точки зрения требований к протоколу MQTT, где x.x.x — номер раздела, y — порядковый номер положения в рамках раздела. Обозначение размещается непосредственно после первого появления соответствующего положения в тексте настоящего стандарта.

3.1.2 **U+XXXX:** Обозначает номер символа в кодировке UTF-8, где X — символ числа в 16-м исчислении (от 0 до F).

3.2 Сокращения

MQTT Клиент-серверный протокол передачи сообщений, основанный на модели «издатель — подписчик»

UTF-8 Формат преобразования Юникода, 8-бит

ASCII 7-битный кодированный набор символов

QoS Качество предоставляемых услуг

TCP/IP Набор сетевых протоколов передачи данных

UDP Протокол пользовательских датаграмм

TLS Протокол защиты транспортного уровня

IANA Администрация адресного пространства Интернет

SSL Уровень защищенных сокетов

4 Представление данных

4.1 Бит

Биты в байте обозначены с 7 по 0. Бит номер 7 является самым значимым битом, наименее значащему биту присваивается номер 0.

4.2 Целочисленные значения данных

Целочисленные значения данных составляют 16 бит в обратном порядке байтов: старший байт предшествует младшему байту.

Это означает, что 16-битное слово представлено в сети как наиболее значащий байт (MSB), за которым следует наименее значащий байт (LSB).

4.3 Закодированные строки UTF-8

Текстовые поля в управляющих пакетах, описанных ниже, кодируются как строки UTF-8. UTF-8 [2] — эффективный метод кодирования символов Unicode [5], который оптимизирует кодовую таблицу ASCII для поддержки обмена текстовыми сообщениями.

Каждой из этих строк предшествует префикс длиной в два байта, который определяет количество байтов в самой кодированной строке UTF-8, как приведено в таблице 1. Следовательно, существует ограничение на размер строки, которая может быть передана в одном из этих кодированных компонентов UTF-8; невозможно передать строку размером более 65535 байт.

Если не указано иное, все кодированные строки UTF-8 могут иметь любую длину в диапазоне от 0 до 65535 байт.

Таблица 1 — Структура кодированных строк UTF-8

Бит	7	6	5	4	3	2	1	0
Байт 1	Длина строки MSB							
Байт 2	Длина строки LSB							
Байт 3...	Данные закодированного символа UTF-8, если длина > 0							

Символьные данные в кодированной строке UTF-8 ДОЛЖНЫ быть правильно сформированы, как определено спецификацией Unicode [5] и подтверждено в RFC 3629 [2]. В частности, эти данные НЕ ДОЛЖНЫ включать кодовые точки между U+D800 и U+DFFF. Если Сервер или Клиент получает управляющий пакет, содержащий неправильно сформированный UTF-8, он ДОЛЖЕН прервать сетевое подключение [MQTT-4.5.3-1].

Закодированная строка UTF-8 НЕ ДОЛЖНА включать кодировку нулевого символа U+0000. Если получатель (Сервер или Клиент) получает управляющий пакет, содержащий U+0000, он ДОЛЖЕН прервать сетевое соединение [MQTT-4.5.3-2].

В данные НЕ СЛЕДУЕТ включать кодовые пункты Unicode [5], перечисленные ниже. Если получатель (Сервер или Клиент) получает управляющий пакет, содержащий любой из них, он МОЖЕТ прервать сетевое соединение:

- управляющие символы U+0001..U+001F;
- управляющие символы U+007F..U+009F;
- кодовые точки, которые согласно спецификации Unicode [5] не являются символами (например, U+0FFFF).

Кодированная UTF-8 последовательность 0xEF 0xBB 0xBF всегда должна интерпретироваться как U+FEFF («НЕРАЗРЫВНЫЙ ПРОБЕЛ С НУЛЕВОЙ ШИРИНОЙ»), где бы она ни появлялась в строке, и НЕ ДОЛЖНА быть пропущена или удалена получателем пакета [MQTT-1.5.3-3].

Пример — Строка А, которая представлена ЗАГЛАВНОЙ ЛАТИНСКОЙ БУКВОЙ А, за которой следует кодовая точка U+2A6D4 (представляющей ИДЕОГРАФИЧЕСКОЕ РАСШИРЕНИЕ CJK символ B), кодируется следующим образом, приведенным в таблице 2.

Таблица 2 — Пример строки в кодировке UTF-8

Бит	7	6	5	4	3	2	1	0
Байт 1	Длина строки MSB (0x00)							
	0	0	0	0	0	0	0	0
Байт 2	Длина строки LSB (0x05)							
	0	0	0	0	0	1	0	1
Байт 3	‘A’ (0x41)							
	0	1	0	0	0	0	0	1
Байт 4	(0xF0)							
	1	1	1	1	0	0	0	0
Байт 5	(0xAA)							
	1	0	1	0	1	0	1	0
Байт 6	(0x9B)							
	1	0	0	1	1	0	1	1
Байт 7	(0x94)							
	1	0	0	1	0	1	0	0

5 Формат управляющего пакета MQTT

5.1 Структура управляющего пакета MQTT

Протокол MQTT работает, обмениваясь серией управляющих пакетов MQTT определенным образом. В данном разделе описывается формат таких пакетов.

Управляющий пакет MQTT состоит из трех частей, всегда расположенных в следующем порядке, приведенном в таблице 3.

Таблица 3 — Структура управляющего пакета MQTT

№	Части управляющего пакета MQTT
1	Фиксированный заголовок, присутствующий во всех пакетах управления MQTT
2	Переменный заголовок, присутствующий в некоторых пакетах управления MQTT
3	Полезная нагрузка, присутствующая в некоторых пакетах управления MQTT

5.2 Фиксированный заголовок

Каждый управляющий пакет MQTT содержит фиксированный заголовок. В таблице 4 приведен фиксированный формат заголовка.

Таблица 4 — Фиксированный формат заголовка

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT				Флаги, определенные для каждого типа управляющего пакета MQTT			
Байт 2...	Остаток байтов							

5.2.1 Тип управляющего пакета MQTT

Позиция: байт 1, биты 7-4.

Значения, представленные в виде 4-разрядной величины без знака, приведены в таблице 5.

Таблица 5 — Типы управляющих пакетов

Название	Значение	Направление потока	Описание
Зарезервировано	0	Запрещено	Зарезервировано
CONNECT	1	Клиент — Сервер	Запрос Клиента на подключение к Серверу
CONNACK	2	Сервер — Клиент	Подтверждение подключения
PUBLISH	3	Клиент — Сервер или Сервер — Клиент	Опубликовать сообщение
PUBACK	4	Клиент — Сервер или Сервер — Клиент	Подтверждение публикации
PUBREC	5	Клиент — Сервер или Сервер — Клиент	Получение публикации (часть1 механизма гарантированной поставки)
PUBREL	6	Клиент — Сервер или Сервер — Клиент	Сообщение для публикации отправлено (часть 2 механизма гарантированной поставки
PUBCOMP	7	Клиент — Сервер или Сервер — Клиент	Публикация сообщения окончена (часть 3 механизма гарантированной поставки)
SUBSCRIBE	8	Клиент — Сервер	Запрос подписки от Клиента
SUBACK	9	Сервер — Клиент	Подтверждение подписки
UNSUBSCRIBE	10	Клиент — Сервер	Запрос об отказе от подписки
UNSUBACK	11	Сервер — Клиент	Подтверждение отказа от подписки
PINGREQ	12	Клиент — Сервер	Запрос PING
PINGRESP	13	Сервер — Клиент	Ответ PING
DISCONNECT	14	Клиент — Сервер	Сообщение об отключении Клиента
Зарезервировано	15	Запрещено	Зарезервировано

5.2.2 Флаги

Остальные биты [3-0] байта 1 в фиксированном заголовке содержат флаги, специфичные для каждого типа управляющего пакета MQTT, как приведено в таблице 6. Если в таблице 6 бит флага отмечен, как «Зарезервировано», то он зарезервирован для будущего использования и ДОЛЖЕН быть установлен в значение, указанное в данной таблице [MQTT-5.2.2-1]. Если полученный пакет содержит неверные флаги, получатель ДОЛЖЕН прервать сетевое соединение [MQTT-5.2.2-2]. Более подробная информация об обработке ошибок приведена в подразделе 7.8.

Таблица 6 — Биты флагов

Управляющий пакет	Фиксированные флаги заголовков	Бит 3	Бит 2	Бит 1	Бит 0
CONNECT	Зарезервировано	0	0	0	0
CONNACK	Зарезервировано	0	0	0	0
PUBLISH	Используется в MQTT 3.1.1	DUP[1]	QoS[2]	QoS[2]	RETAIN[3]
PUBACK	Зарезервировано	0	0	0	0
PUBREC	Зарезервировано	0	0	0	0
PUBREL	Зарезервировано	0	0	1	0
PUBCOMP	Зарезервировано	0	0	0	0
SUBSCRIBE	Зарезервировано	0	0	1	0
SUBACK	Зарезервировано	0	0	0	0
UNSUBSCRIBE	Зарезервировано	0	0	1	0
UNSUBACK	Зарезервировано	0	0	0	0
PINGREQ	Зарезервировано	0	0	0	0
PINGRESP	Зарезервировано	0	0	0	0
DISCONNECT	Зарезервировано	0	0	0	0

DUP[1] = Повторная доставка управляющего пакета PUBLISH
QoS[2] = Качество услуги передачи PUBLISH
RETAIN[3] = Флаг сохранения пакета PUBLISH
В пункте 6.3.1 приведено описание DUP, QoS и RETAIN в составе управляющего пакета PUBLISH.

5.2.3 Остаток байтов

Позиция: начинается с байта 2.

Остаток байтов — это количество байтов, оставшихся в текущем пакете, включая данные в переменном заголовке и полезную нагрузку. В это число не входят байты, используемые для кодирования самого числа оставшихся байтов, указанного в фиксированном заголовке.

Число оставшихся в данном пакете байтов кодируется с использованием схемы кодирования с переменной длиной, которая использует один байт для значений до 127. Более крупные величины обрабатываются следующим образом. Наименее значимые семь бит каждого байта кодируют данные, а самый старший бит используется для указания, что в представлении присутствуют следующие байты. Таким образом, каждый байт кодирует 128 значений и «бит продолжения». Максимальное количество байтов в поле «Остаток байтов» равно четырем.

Пример — Десятичное число 64 кодируется как один байт, десятичное значение 64, шестнадцатеричное 0x40. Десятичное число 321 (= 65 + 2*128) кодируется как два байта, наименее значимые указываются первыми. Первый байт равен 65 + 128 = 193. Обратите внимание, что верхний бит установлен для указания хотя бы одного следующего байта. Второй байт равен 2.

Примечание — Кодирование остатка байтов позволяет приложениям отправлять управляющие пакеты размером до 268435455 (256 МБ). Представление этого числа в закодированном виде: 0xFF, 0xFF, 0xFF, 0x7F.

В таблице 7 приведены значения остатка байтов, представленные в порядке возрастания числа байтов.

Таблица 7 — Размер поля остатка байтов

Цифры	От	До
1	0 (0x00)	127 (0x7F)
2	128 (0x80, 0x01)	16 383 (0xFF, 0x7F)
3	16 384 (0x80, 0x80, 0x01)	2 097 151 (0xFF, 0xFF, 0x7F)
4	2 097 152 (0x80, 0x80, 0x80, 0x01)	268 435 455 (0xFF, 0xFF, 0xFF, 0x7F)

Примечания

1 Алгоритм кодирования неотрицательного целого (X) в схему кодирования с переменной длиной выглядит следующим образом:

- ввод
- encodedByte = X MOD 128
- X = X DIV 128
- // если для кодирования данных доступно больше данных, установите верхний бит этого байта
- если (X > 0)
- encodedByte = encodedByte ИЛИ 128
- endif
- 'output' encodedByte
- когда (X > 0)

если MOD является оператором modulo (% in C), DIV является целым делением (/ in C), а OR — поразрядным или (| в C).

2 Алгоритм декодирования поля «Остаток байтов» выглядит следующим образом:

- множитель = 1
- значение = 0
- ввод
- encodedByte = «следующий байт из потока»
- значение + = (encodedByte AND 127) * множитель
- множитель * = 128
- если (множитель> 128 * 128 * 128)
- выводит ошибку (недопустимая Остаток байтов)
- когда ((encodedByte AND 128)! = 0)
- где AND — бит-бит и оператор (& в C).

Когда этот алгоритм завершается, величина содержит значение «Оставшейся длины».

5.3 Переменный заголовок

Некоторые типы управляющих пакетов MQTT содержат компонент переменного заголовка. Он находится между фиксированным заголовком и полезной нагрузкой. Содержимое переменного заголовка зависит от типа управляющего пакета. Поле «Идентификатор пакета» переменного заголовка является общим для нескольких типов пакетов.

5.3.1 Идентификатор пакета

В таблице 8 приведены байты идентификатора пакета.

Таблица 8 — Байты идентификатора пакета

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

Компонент переменного заголовка для многих типов управляющих пакетов включает в себя поле идентификатора пакета из 2 байтов. Такими управляющими пакетами являются PUBLISH (где QoS > 0), PUBACK, PUBREC, PUBREL, PUBCOMP, SUBSCRIBE, SUBACK, UNSUBSCRIBE, UNSUBACK.

Управляющие пакеты SUBSCRIBE, UNSUBSCRIBE и PUBLISH (в случаях, когда QoS > 0) ДОЛЖНЫ содержать ненулевой 16-разрядный идентификатор пакета [MQTT-5.3.1-1]. Каждый раз, когда Клиент отправляет новый пакет одного из этих типов, он ДОЛЖЕН назначить ему неиспользуемый в настоящее

время идентификатор пакета [MQTT-5.3.1-2]. Если Клиент повторно отправляет конкретный управляющий пакет, он ДОЛЖЕН использовать тот же идентификатор пакета в последующих повторных отправках этого пакета. Идентификатор пакета становится доступным для повторного использования после того, как Клиент обработал соответствующий пакет подтверждения. В случае QoS 1 PUBLISH — это соответствующий PUBACK; в случае QoS 2 — это PUBCOMP. Для SUBSCRIBE или UNSUBSCRIBE — это соответствующий SUBACK или UNSUBACK [MQTT-5.3.1-3]. Те же условия применяются к Серверу, когда он отправляет PUBLISH с QoS > 0 [MQTT-5.3.1-4].

Пакет PUBLISH НЕ ДОЛЖЕН содержать идентификатор пакета, если его значение QoS установлено на 0 [MQTT-5.3.1-5].

Пакеты PUBACK, PUBREC или PUBREL ДОЛЖНЫ содержать тот же идентификатор пакета, что и первоначально отправленный пакет PUBLISH [MQTT-5.3.1-6]. Аналогично SUBACK и UNSUBACK ДОЛЖНЫ содержать идентификатор пакета, который использовался в соответствующем пакете SUBSCRIBE и UNSUBSCIBE, соответственно [MQTT-5.3.1-7].

Управляющие пакеты, требующие идентификатор пакета, приведены в таблице 9.

Таблица 9 — Управляющие пакеты, содержащие идентификатор пакета

Управляющий пакет	Поле идентификатора пакета
CONNECT	НЕТ
CONNACK	НЕТ
PUBLISH	ДА (если QoS > 0)
PUBACK	ДА
PUBREC	ДА
PUBREL	ДА
PUBCOMP	ДА
SUBSCRIBE	ДА
SUBACK	ДА
UNSUBSCRIBE	ДА
UNSUBACK	ДА
PINGREQ	НЕТ
PINGRESP	НЕТ
DISCONNECT	НЕТ

Клиент и Сервер назначают идентификаторы пакетов независимо друг от друга. В результате пары Клиент-Сервер могут участвовать в параллельных обменах сообщений с использованием одинаковых идентификаторов пакетов.

Пример — Клиент может отправить пакет PUBLISH с идентификатором пакета 0x1234, а затем получить другой пакет PUBLISH с идентификатором пакета 0x1234 со своего Сервера, прежде чем он получит PUBACK по отправленному PUBLISH.

Клиент	Сервер
Идентификатор пакета PUBLISH = 0x1234 --- →	
← -- Идентификатор пакета PUBLISH = 0x1234	
Идентификатор пакета PUBACK = 0x1234 --- →	
← -- Идентификатор пакета PUBACK = 0x1234	

5.4 Полезная нагрузка

Некоторые управляющие пакеты MQTT содержат полезную нагрузку в качестве конечной части пакета, как описано в главе 6. В случае пакета PUBLISH — это сообщение приложения. В таблице 10 приведены управляющие пакеты, в которые необходимо включать полезную нагрузку.

Таблица 10 — Управляющие пакеты, содержащие полезную нагрузку

Управляющий пакет	Полезная нагрузка
CONNECT	Обязательно
CONNACK	Нет
PUBLISH	Опционально
PUBACK	Нет
PUBREC	Нет
PUBREL	Нет
PUBCOMP	Нет
SUBSCRIBE	Обязательно
SUBACK	Обязательно
UNSUBSCRIBE	Обязательно
UNSUBACK	Нет
PINGREQ	Нет
PINGRESP	Нет
DISCONNECT	Нет

6 Управляющие пакеты MQTT

6.1 CONNECT — Клиент запрашивает подключение к Серверу

После того, как Клиент установил сетевое соединение с Сервером, первым пакетом, отправленным Клиентом на Сервер, ДОЛЖЕН быть пакет CONNECT (СОЕДИНЕНИЕ) [MQTT-6.1.0-1].

Клиент может отправить пакет CONNECT через сетевое подключение только один раз. Сервер ДОЛЖЕН обработать второй пакет CONNECT, отправленный Клиентом, как нарушение протокола, и прервать соединение с Клиентом [MQTT-6.1.0-2]. Информация об обработке приведена в подразделе 7.8.

Полезная нагрузка содержит одно или несколько закодированных полей. Они определяют уникальный идентификатор Клиента для Клиента, тему «последней воли», сообщение «последней воли», имя пользователя и пароль. Все поля, кроме идентификатора Клиента, являются необязательными и их наличие определяется на основе флагов в переменном заголовке.

6.1.1 Фиксированный заголовок

В таблице 11 приведен фиксированный заголовок пакета CONNECT.

Таблица 11 — Фиксированный заголовок пакета CONNECT

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (1)				Зарезервировано			
	0	0	0	1	0	0	0	0
Байт 2...	Остаток байтов							

Поле «Остаток байтов»

Остаток байтов — это длина переменного заголовка (10 байт) плюс длина полезной нагрузки. Суммарное количество байт в остатке кодируется способом, описанным в подпункте 5.2.3.

6.1.2 Переменный заголовок

Переменный заголовок для пакета CONNECT состоит из четырех полей в следующем порядке: имя протокола, уровень протокола, флаги подключения и интервала поддержки активного соединения.

6.1.2.1 Имя протокола

В таблице 12 приведены байты имени протокола.

Таблица 12 — Байты имени протокола

Имя протокола	Описание	7	6	5	4	3	2	1	0
Байт 1	Длина MSB (0)	0	0	0	0	0	0	0	0
Байт 2	Длина LSB (4)	0	0	0	0	0	1	0	0
Байт 3	«M»	0	1	0	0	1	1	0	1
Байт 4	«Q»	0	1	0	1	0	0	0	1
Байт 5	«T»	0	1	0	1	0	1	0	0
Байт 6	«T»	0	1	0	1	0	1	0	0

Имя протокола — это кодированная строка UTF-8, которая представляет собой имя протокола «MQTT», записанное заглавными буквами, как показано. Строка, ее смещение и длина не меняются в последующих изданиях спецификации MQTT.

Если имя протокола неверно, Сервер МОЖЕТ прервать соединение с Клиентом, или МОЖЕТ продолжить обработку пакета CONNECT в соответствии с некоторыми другими спецификациями. В последнем случае Сервер НЕ ДОЛЖЕН продолжать обрабатывать пакет CONNECT в соответствии с настоящей спецификацией [MQTT-6.1.2-1].

Пример — Инспекторы пакетов, например, брандмауэры, могут использовать Имя протокола для идентификации трафика MQTT.

6.1.2.2 Уровень протокола

В таблице 13 приведен байт уровня протокола.

Таблица 13 — Байт уровня протокола

Уровень протокола	Описание	7	6	5	4	3	2	1	0
Байт 7	Уровень (4)	0	0	0	0	0	1	0	0

8-битная величина без знака, представляющая уровень версии протокола, используемый Клиентом. Значение поля «Уровень протокола» для версии 3.1.1 протокола 4 (0x04). Сервер ДОЛЖЕН ответить на пакет CONNECT передачей пакета CONNACK с кодом 0x01 (неподдерживаемая версия протокола), а затем прервать соединение с Клиентом, если уровень протокола не поддерживается Сервером [MQTT-6.1.2-2].

6.1.2.3 Флаги соединения

Байт флагов соединения содержит ряд параметров, определяющих поведение соединения MQTT. В нем также указывается наличие или отсутствие полей в полезной нагрузке. В таблице 14 приведены байты флагов соединения.

Таблица 14 — Байты флагов соединения

Бит	7	6	5	4	3	2	1	0
	Флаг имени пользователя	Флаг пароля	Флаг сохранения «последней воли»	QoS «последней воли»		Флаг «последней воли»	Флаг очистки Сеанса	Зарезервировано
Байт 8	X	X	X	X	X	X	X	X

Сервер ДОЛЖЕН проверить, что зарезервированный флаг в пакете управления CONNECT установлен на ноль и прервать соединение с Клиентом, если значение отлично от нуля [MQTT-6.1.2-3].

6.1.2.4 Флаг очистки Сеанса

Позиция: бит 1 байта флага соединения.

Значение этого бита определяет способ обработки состояния Сеанса.

Клиент и Сервер могут сохранять информацию о состоянии Сеанса, чтобы обеспечить надежную передачу сообщений в последовательности сетевых подключений. Данный бит используется для управления временем хранения состояния Сеанса.

Если для флага очистки Сеанса установлено значение «0», Сервер ДОЛЖЕН возобновить обмен данными с Клиентом на основе состояния текущего сеанса (как определено идентификатором Клиента). Если нет Сеанса, связанного с идентификатором Клиента, Сервер ДОЛЖЕН создать новый сеанс. Клиент и Сервер ДОЛЖНЫ сохранять информацию о состоянии текущего Сеанса после прекращения соединения между Клиентом и Сервером [MQTT-6.1.2-4]. После завершения Сеанса, по которому параметр «Очистить сеанс» установлен на 0, Сервер ДОЛЖЕН сохранить дополнительные сообщения с уровнем качества обслуживания 1 и 2, соответствующие любым подпискам, которые существовали у Клиента во время завершения Сеанса [MQTT-6.1.2- 5]. Он МОЖЕТ также хранить сообщения с уровнем качества обслуживания 0, соответствующие тем же критериям.

Если для флага очистки Сеанса установлено значение «1», Клиент и Сервер ДОЛЖНЫ завершить любой предыдущий Сеанс между Клиентом и Сервером и запустить новый. Этот Сеанс длится до тех пор, пока существует сетевое подключение. Данные состояния, связанные с этим Сеансом, НЕ ДОЛЖНЫ повторно использоваться в любом последующем сеансе [MQTT-6.1.2-6].

Состояние сеанса у Клиента состоит из:

- сообщений с уровнем качества обслуживания 1 и 2, которые были отправлены на Сервер, но не были полностью подтверждены;
- сообщений с уровнем качества обслуживания 2, которые были получены с Сервера, но не были полностью подтверждены.

Состояние Сеанса на Сервере состоит из:

- признака существования Сеанса, даже если остальная часть состояния Сеанса пуста;
- подписок Клиента;
- QoS 1 и QoS 2, которые были отправлены Клиенту, но не были полностью подтверждены.
- QoS 1 и QoS 2, ожидающие передачи Клиенту;
- сообщения QoS 2, полученные от Клиента, но не полностью подтвержденные;
- опционально сообщения QoS 0, ожидающие передачи Клиенту.

Сохраненные сообщения не являются частью состояния Сеанса на Сервере, они НЕ ДОЛЖНЫ удаляться, когда Сеанс заканчивается [MQTT-6.1.2.7].

Более подробная информация и ограничения сохраненного состояния приведены в подразделе 4.1.

Если для параметра «Очистить сеанс» установлено значение 1, Клиент и Сервер не должны обрабатывать удаление состояния атомарно.

Примечания

1 Чтобы обеспечить надлежащее состояние в случае сбоя связи, Клиент должен повторить попытки подключения, установив параметр «Очистить сеанс» на 1, пока соединение не будет успешно выполнено.

2 Как правило, Клиент всегда будет подключаться с использованием параметра «Очистить Сеанс», установленного на 0 или на 1, и не будет переключаться с одного значения на другое. Выбор будет зависеть от приложения. Клиент, использующий «Очистить Сеанс», установленный на 1, не получит старые сообщения приложения и должен заново подписываться на любые темы, которые его интересуют, при каждом подключении. Клиент, использующий «Очистить Сеанс», установленный на 0, получит все сообщения QoS 1 или QoS 2, которые были опубликованы, пока он был отключен. Следовательно, чтобы гарантировать получение сообщений после отключения, используйте QoS 1 или QoS 2 с параметром «Очистить Сеанс», установленным на 0.

3 При подключении Клиента с параметром «Очистить сеанс», установленным на 0, он запрашивает, чтобы Сервер сохранил состояние сеанса MQTT после его отключения. Клиенты должны подключаться только с параметром «Очистить Сеанс», установленным на 0, если они планируют повторно подключиться к Серверу в какой-то более поздний момент времени. Когда Клиент решил, что он больше не использует Сеанс, он должен выполнить окончательное соединение с параметром «Очистить Сеанс», установленным на 1, а затем отключиться.

6.1.2.5 Флаг «последней воли»

Позиция: 2 бита флагов подключения.

Установка флага «последней воли» на 1 означает, что, если запрос о соединении принят, сообщение «последней воли» ДОЛЖНО храниться на Сервере и быть связано с сетевым подключением. Сообщение о «последней воле» ДОЛЖНО быть опубликовано, когда Сетевое подключение будет закрыто, если сообщение не было удалено Сервером при получении пакета DISCONNECT [MQTT-6.1.2-8].

Ситуации, в которых публикуется сообщение о дальнейших действиях, включают, но не ограничиваются:

- ошибкой ввода-вывода или сетевым сбоем, обнаруженными Сервером;
- Клиент не может выполнить соединение в течение времени активного соединения;
- Клиент закрывает сетевое соединение без предварительной отправки пакета DISCONNECT;
- Сервер закрывает сетевое подключение из-за ошибки протокола.

Если флаг «последней воли» установлен на 1, поля уровня качества обслуживания сообщения «последней воли» и сохранение сообщения «последней воли» во флагах соединения будут использоваться Сервером, а поля темы «последней воли» и сообщения «последней воли» ДОЛЖНЫ присутствовать в полезной нагрузке [MQTT-6.1.2-9].

Сообщение о «последней воле» ДОЛЖНО быть удалено из сохраненного состояния Сеанса на Сервере после публикации, или когда Сервер получил от Клиента пакет DISCONNECT [MQTT-6.1.2-10].

Если флаг дальнейших действий установлен на 0, значения полей QoS и будущее сохранение должны быть установлены на ноль, а поля будущей темы и сообщение о дальнейших действиях НЕ ДОЛЖНЫ присутствовать в полезной нагрузке [MQTT-6.1.2-11].

Если флаг дальнейших действий установлен на 0, сообщение «последней воли» НЕ ДОЛЖНО публиковаться при завершении данного сетевого подключения [MQTT-6.1.2-12].

Серверу СЛЕДУЕТ немедленно публиковать сообщения «последней воли». В случае отключения или сбоя Сервера, он МОЖЕТ отложить публикацию сообщений «последней воли» до последующего перезапуска. Если это произойдет, может наблюдаться задержка между временем, когда произошел сбой Сервера, и временем публикации сообщений «последней воле».

6.1.2.6 Уровень качества обслуживания (QoS) «последней воли»

Позиция: биты 4 и 3 флагов соединения.

Эти два бита определяют уровень качества обслуживания QoS, который будет использоваться при публикации сообщений «последней воли».

Если флаг «последней воли» установлен на 0, то значение QoS «последней воли» ДОЛЖНО быть установлено на 0 (0x00) [MQTT-6.1.2-13].

Если флаг «последней воли» установлен на 1, значение QoS «последней воли» может быть 0 (0x00), 1 (0x01) или 2 (0x02). Значение НЕ ДОЛЖНО быть установлено на 3 (0x03) [MQTT-6.1.2-14].

6.1.2.7 Флаг сохранения «последней воли»

Позиция: бит 5 флагов соединения.

Этот бит указывает, следует ли сохранять сообщение «последней воли», когда оно опубликовано.

Если флаг «последней воли» установлен на 0, то флаг будущего сохранения ДОЛЖЕН быть установлен на 0 [MQTT-6.1.2-15].

Флаг «последней воли» установлен на 1 в следующих случаях:

- если для будущего сохранения установлено значение 0, Сервер ДОЛЖЕН публиковать сообщение о дальнейших действиях как не сохраненное сообщение [MQTT-6.1.2-16];
- если для будущего сохранения установлено значение 1, Сервер ДОЛЖЕН публиковать сообщение о дальнейших действиях как сохраненное сообщение [MQTT-6.1.2-17].

6.1.2.8 Флаг имени пользователя

Позиция: бит 7 флагов соединения.

Если параметр флага имени пользователя установлен на 0, имя пользователя НЕ ДОЛЖНО присутствовать в полезной нагрузке [MQTT-6.1.2-18].

Если параметр флага имени пользователя установлен на 1, имя пользователя ДОЛЖНО присутствовать в полезной нагрузке [MQTT-6.1.2-19].

6.1.2.9 Флаг пароля

Позиция: бит 6 байтов флагов соединения.

Если флаг пароля установлен на 0, пароль НЕ ДОЛЖЕН присутствовать в полезной нагрузке [MQTT-6.1.2-20].

Если флаг пароля установлен на 1, пароль ДОЛЖЕН присутствовать в полезной нагрузке [MQTT-6.1.2-21].

Если параметр флага имени пользователя установлен на 0, флаг пароля ДОЛЖЕН быть установлен на 0 [MQTT-6.1.2-22].

6.1.2.10 Интервал поддержки активного соединения

В таблице 15 приведены байты активного соединения.

Таблица 15 — Байты активного соединения

Бит	7	6	5	4	3	2	1	0
Байт 9	Активное соединение MSB							
Байт 10	Активное соединение LSB							

Активное соединение — это временной интервал, измеряемый в секундах и представляемый в виде 16-битного слова, что является максимально допустимым временным промежутком между моментом, в который Клиент заканчивает передачу одного управляющего пакета и моментом, в который он начинает отправлять следующий. Клиент несет ответственность за то, чтобы промежуток между отправляемыми управляющими пакетами не превышал значение активного соединения. В случае отсутствия необходимости в отправке каких-либо других управляющих пакетов Клиент ДОЛЖЕН отправить пакет PINGREQ [MQTT-6.1.2-23].

Клиент может отправить PINGREQ в любое время, независимо от значения активного соединения, и использовать PINGRESP для определения того, что сеть и Сервер работают.

Если значение активного соединения не равно нулю, и Сервер не получает управляющий пакет от Клиента в течение полутора интервалов периода активного соединения, он ДОЛЖЕН отключить сетевое соединение с Клиентом, как если бы произошел сбой сети [MQTT -3.1.2-24].

Если Клиент не получает пакет PINGRESP в течение разумного промежутка времени после отправки PINGREQ, ему СЛЕДУЕТ прервать сетевое подключение к Серверу.

Значение активного соединения, равное нулю (0), приводит к отключению механизма отслеживания активного соединения. Это означает, что в этом случае Сервер не обязан отключать Клиента по причине отсутствия действий.

Обратите внимание, что Серверу разрешено отключать Клиента, которого он определяет, как неактивного или не реагирующего, в любое время, независимо от значения активного соединения, предоставленного этим Клиентом.

Примечание — Фактическое значение активного соединения является специфичным для приложения; обычно оно составляет несколько минут. Максимальное значение составляет 18 часов 12 минут и 15 секунд.

Пример — Переменный заголовок приведен в таблице 16.

Таблица 16 — Пример переменного заголовка

	Описание	7	6	5	4	3	2	1	0
Имя протокола									
Байт 1	Длина MSB (0)	0	0	0	0	0	0	0	0
Байт 2	Длина LSB (4)	0	0	0	0	0	1	0	0
Байт 3	«M»	0	1	0	0	1	1	0	1
Байт 4	«Q»	0	1	0	1	0	0	0	1
Байт 5	«T»	0	1	0	1	0	1	0	0
Байт 6	«T»	0	1	0	1	0	1	0	0
Уровень протокола									
Байт 7	Уровень (4)	0	0	0	0	0	1	0	0

Окончание таблицы 16

	Описание	7	6	5	4	3	2	1	0
Флаги соединения									
Байт 8	Флаг с именем пользователя (1) Флаг пароля (1) Будущее сохранение (0) Будущее QoS (01) Флаг дальнейших действий (1) Очистить сеанс (1) Зарезервировано (0)	1	1	0	0	1	1	1	0
Активное соединение									
Байт 9	Активное соединение MSB (0)	0	0	0	0	0	0	0	0
Байт 10	Активное соединение LSB (10)	0	0	0	0	1	0	1	0

6.1.3 Полезная нагрузка

Полезная нагрузка пакета CONNECT содержит одно или несколько полей с префиксом длины, наличие которых определяется флагами в переменном заголовке. Эти поля, если они есть, ДОЛЖНЫ появляться в порядке: идентификатор Клиента, тема «последней воли», сообщение «последней воли», имя пользователя, пароль [MQTT-6.1.3-1].

6.1.3.1 Идентификатор Клиента

Идентификатор Клиента (ClientId) позволяет Серверу идентифицировать Клиента. Каждый Клиент, подключающийся к Серверу, имеет уникальный Идентификатор Клиента (ClientId). Идентификатор Клиента ДОЛЖЕН использоваться Клиентами и Серверами для определения их состояния в отношении этого сеанса MQTT между Клиентом и Сервером [MQTT-6.1.3-2].

Идентификатор Клиента (ClientId) ДОЛЖЕН присутствовать и ДОЛЖЕН быть первым полем в полезной нагрузке пакета CONNECT [MQTT-6.1.3-3].

ClientId ДОЛЖЕН быть кодированной строкой UTF-8, как определено в разделе 4.5.3 [MQTT-6.1.3-4].

Сервер ДОЛЖЕН обрабатывать идентификаторы Клиентов, длина которых попадает в диапазон 1—23 кодированных байтов UTF-8, и которые состоят только из символов «0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ» [MQTT-6.1.3-5].

Сервер МОЖЕТ обрабатывать идентификаторы Клиентов, которые содержат более 23 закодированных байтов. Сервер МОЖЕТ разрешить ClientId, содержащие символы, не включенные в список, указанный выше.

Сервер МОЖЕТ позволить Клиенту предоставить ClientId, длина которого равна нулю, однако, если он делает это, Сервер ДОЛЖЕН рассматривать это как особый случай и назначить уникальный ClientId для этого Клиента. Он ДОЛЖЕН обрабатывать пакет CONNECT, как если бы Клиент предоставил уникальный ClientId [MQTT-6.1.3-6].

Если Клиент отправляет ClientId с нулевым байтом, Клиент ДОЛЖЕН также установить флаг очистки Сеанса на 1 [MQTT-6.1.3-7].

Если Клиент отправляет ClientId с нулевым байтом, с установленным значением флага очистки Сеанса равным 0, Сервер ДОЛЖЕН отвечать на управляющий пакет CONNECT отправкой пакета CONNACK с кодом возврата 0x02 (отклонение идентификатора), а затем прервать сетевое соединение [MQTT-6.1.3-8].

Если Сервер отклоняет ClientId, он ДОЛЖЕН отвечать на пакет CONNECT отправкой CONNACK с кодом возврата 0x02 (идентификатор отклонен), а затем прервать сетевое соединение [MQTT-6.1.3-9].

Примечание — Реализация Клиента может обеспечить удобный метод для генерации случайного ClientId. Использование такого метода не рекомендуется, если для параметра «Очистить сеанс» установлено значение 0.

6.1.3.2 Тема «последней воли»

Если флаг «последней воли» установлен на 1, тема «последней воли» будет следующим полем в полезной нагрузке. Тема «последней воли» ДОЛЖНА быть кодированной строкой UTF-8, как определено в Разделе 4.5.3 [MQTT-6.1.3-10].

6.1.3.3 Сообщение «последней воли»

Если флаг «последней воли» установлен на 1, то это будет следующее поле в полезной нагрузке. Сообщение «последней воли» определяет сообщение приложения, которое должно быть опубликовано в теме, указанной в поле Тема «последней воли», как описано в подпункте 6.1.2.5. Это поле состоит из двух байт, определяющих размер сообщения, за которым следует полезная нагрузка для будущего сообщения, выраженная в виде последовательности из нуля или более байтов. Поле размера сообщения определяет количество байтов в следующих ниже данных и не включает в себя 2 байта, занятых самим полем.

Когда сообщение «последней воли» публикуется в теме «последней воли», его полезная нагрузка состоит из данных сообщения, и не включает поле размера сообщения.

6.1.3.4 Имя пользователя

Если флаг имени пользователя установлен на 1, является следующим в полезной нагрузке. Имя пользователя ДОЛЖНО быть кодированной строкой UTF-8, как определено в пункте 4.5.3 [MQTT-6.1.3-11]. Оно может использоваться Сервером для аутентификации и авторизации.

6.1.3.5 Пароль

Если флаг пароля установлен на 1, поле, определяющее пароль пользователя, является следующим в полезной нагрузке. Поле «Пароль» содержит от 0 до 65535 байтов двоичных данных с двухбайтовым префиксом, указывающим количество байтов, занятых двоичными данными (оно не включает два байта, занятых самим префиксом). Байты пароля приведены в таблице 17.

Таблица 17 — Байты пароля

Бит	7	6	5	4	3	2	1	0
Байт 1	Размер данных MSB							
Байт 2	Размер данных LSB							
Байт 3…	Данные, если длина > 0							

6.1.4 Ответ

Важным моментом является то, что Сервер МОЖЕТ поддерживать несколько протоколов (включая более ранние версии этого протокола) на одном и том же TCP-порту или другой конечной точке сети. Если Сервер определяет, что протокол является MQTT 3.1.1, он проверяет попытку подключения следующим образом:

а) Если Сервер не получает пакет CONNECT в течение разумного промежутка времени после установления сетевого подключения, Серверу СЛЕДУЕТ закрыть соединение.

б) Сервер ДОЛЖЕН подтвердить, что пакет CONNECT соответствует описанию, данному в подразделе 6.1 и закрыть сетевое соединение, не отправляя CONNACK, если он не соответствует [MQTT-6.1.4-1].

в) Сервер МОЖЕТ проверить, чтобы содержимое пакета CONNECT соответствовало любым дополнительным ограничениям и МОЖЕТ выполнять проверки подлинности и авторизации. Если какая-либо из этих проверок не была пройдена, Серверу СЛЕДУЕТ отправить соответствующий пакет CONNACK с ненулевым кодом возврата, как описано в подразделе 6.2, и закрыть сетевое соединение.

Если проверка прошла успешно, Сервер выполняет следующие шаги:

а) Если идентификатор Клиента представляет Клиента, уже подключенного к Серверу, Сервер ДОЛЖЕН прервать соединение с существующим Клиентом [MQTT-6.1.4-2].

б) Сервер ДОЛЖЕН выполнить очистку Сеанса согласно процедуре, описанной в подпункте 6.1.2.4 [MQTT-6.1.4-3].

в) Сервер ДОЛЖЕН подтвердить пакет CONNECT с помощью пакета CONNACK, содержащего нулевой код возврата [MQTT-6.1.4-4].

г) Запустить доставку сообщений и продолжить мониторинг.

Клиентам разрешено отправлять дополнительные управляющие пакеты сразу после отправки пакета CONNECT; Клиентам не нужно дожидаться поступления пакета CONNACK с Сервера. Если Сервер отклоняет CONNECT, он НЕ ДОЛЖЕН обрабатывать любые данные, отправленные Клиентом после пакета CONNECT [MQTT-6.1.4-5].

Примечание — Клиенты обычно ожидают пакет CONNACK. Однако, если Клиент использует возможность отправлять управляющие пакеты, прежде чем он получит CONNACK, это может упростить программную реализацию Клиента, поскольку Клиенту не требуется отслеживать состояние соединения. Клиент соглашается с тем, что любые данные, отправленные им до получения пакета CONNACK с Сервера, не будут обрабатываться, если Сервер отклонит соединение.

6.2 CONNACK — Подтверждение запроса на подключение

Пакет CONNACK — это пакет, отправленный Сервером в ответ на пакет CONNECT, полученный от Клиента. Первый пакет, отправленный с Сервера Клиенту, ДОЛЖЕН быть пакетом CONNACK [MQTT-6.2.0-1].

Если Клиент не получает пакет CONNACK с Сервера в течение разумного промежутка времени, Клиенту СЛЕДУЕТ прервать сетевое соединение. «Разумный» промежуток времени зависит от типа приложения и инфраструктуры связи.

6.2.1 Фиксированный заголовок

Формат фиксированного заголовка приведен в таблице 18.

Таблица 18 — Фиксированный заголовок пакета CONNACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (2)				Зарезервировано			
	0	0	1	0	0	0	0	0
Байт 2	Остаток байтов (2)							
	0	0	0	0	0	0	1	0

Поле «Остаток байтов»

Данное поле определяет размер переменного заголовка. Для пакета CONNACK она имеет значение 2.

6.2.2 Переменный заголовок

Формат переменного заголовка приведен в таблице 19.

Таблица 19 — Переменный заголовок пакета CONNACK

	Описание	7	6	5	4	3	2	1	0
Флаги подтверждения соединения		Зарезервировано							SP[1]
Байт 1		0	0	0	0	0	0	0	X
Код возврата соединения									
Байт 2		X	X	X	X	X	X	X	X

6.2.2.1 Флаги подтверждения соединения

Байт 1 — это «Флаги подтверждения соединения». Биты 7-1 зарезервированы и ДОЛЖНЫ быть установлены на 0.

Бит 0 (SP[1]) — это флаг текущего сеанса.

6.2.2.2 Флаг текущего сеанса

Позиция: бит 0 флагов подтверждения соединения.

Если Сервер принимает соединение с флагом очистки Сеанса, установленным на 1, Сервер ДОЛЖЕН установить флаг текущего Сеанса на 0 в пакете CONNACK в дополнение к установке нулевого кода возврата в пакете CONNACK [MQTT-6.2.2-1].

Если Сервер принимает соединение с флагом очистки Сеанса установленным на 0, значение, устанавливаемое для флага текущего Сеанса, зависит от того, сохранил ли Сервер состояние Сеанса для предоставленного идентификатора Клиента. Если Сервер сохранил состояние сеанса, он ДОЛЖЕН установить флаг текущего Сеанса на 1 в пакете CONNACK [MQTT-6.2.2-2]. Если Сервер не сохранил состояние сеанса, он ДОЛЖЕН установить флаг текущего Сеанса на 0 в пакете CONNACK. Это выполняется дополнительно к установке нулевого кода возврата в пакете CONNACK [MQTT-6.2.2-3].

Флаг текущего Сеанса позволяет Клиенту установить, совпадают ли представления Клиента и Сервера о существовании уже сохраненного состояния Сеанса.

Как только начальная настройка Сеанса завершена, Клиент с сохраненным состоянием Сеанса ожидает, что Сервер будет поддерживать сохраненное состояние Сеанса. В случае, если значение текущего Сеанса, полученного Клиентом с Сервера, не соответствует ожиданиям, Клиент может выбрать, следует ли продолжать Сеанс или необходимо прервать соединение. Клиент может сбросить состояние Сеанса как у Клиента, так и на Сервере, выполнив отключение, установив соединение с флагом очистки сеанса, установленным на 1, а затем снова отключившись.

Если Сервер отправляет пакет CONNACK, содержащий ненулевой код возврата, он ДОЛЖЕН установить флаг текущего Сеанса на 0 [MQTT-6.2.2-4].

6.2.2.3 Подключить Код возврата

Байт 2 в заголовке переменной.

Значения для байтового беззнакового поля кода возврата соединения перечислены в таблице 20. Если Сервер принял правильно сформированный пакет CONNECT, но Сервер не может его обработать по какой-либо причине, тогда Серверу СЛЕДУЕТ попытаться отправить пакет CONNACK, содержащий соответствующий ненулевой код возврата CONNECT из этой таблицы. Если Сервер отправляет пакет CONNACK, содержащий ненулевой код возврата, он ДОЛЖЕН затем прервать сетевое соединение [MQTT-6.2.2-5].

Таблица 20 — Значения кода возврата соединения

Значение	Ответ кода возврата	Описание
0	0x00 Соединение подтверждено	Соединение подтверждено
1	0x01 Соединение отклонено, неприемлемая версия протокола	Сервер не поддерживает уровень протокола MQTT, запрошенный Клиентом
2	0x02 Соединение отклонено, идентификатор отклонен	Идентификатор Клиента — правильно сформированная строка UTF-8, но не разрешен Сервером
3	0x03 Соединение отклонено, Сервер недоступен	Сетевое подключение выполнено, но сервис MQTT недоступен
4	0x04 Соединение отклонено, неправильное имя пользователя или пароль	Данные в имени пользователя или пароле неверны
5	0x05 Соединение отклонено, не разрешено	Клиент не имеет права подключаться
6-255		Зарезервировано для будущего использования

Если ни один из кодов возврата, перечисленных в таблице 20, не считается применимым, тогда Сервер ДОЛЖЕН закрыть сетевое соединение, не отправляя пакет CONNACK [MQTT-6.2.2-6].

6.2.3 Полезная нагрузка

Пакет CONNACK не имеет полезной нагрузки.

6.3 PUBLISH — Публикация сообщения

Управляющий пакет PUBLISH отправляется Клиентом на Сервер или от Сервера к Клиенту для передачи сообщения приложения.

6.3.1 Фиксированный заголовок

В таблице 21 приведен формат фиксированного заголовка.

Таблица 21 — Фиксированный заголовок пакета PUBLISH

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (3)				Флаг DUP	Уровень QoS		Флаг сохранения
	0	0	1	1	X	X	X	X
Байт 2	Остаток байтов							

6.3.1.1 Флаг дублирования (DUP)

Позиция: байт 1, бит 3.

Если для флага дублирования (DUP) установлено значение 0, это означает, что это первый случай, когда Клиент или Сервер попытались отправить этот пакет MQTT PUBLISH. Если флаг дублирования (DUP) установлен на 1, это означает, что это может быть повторной попыткой доставить пакет, уже отправленный ранее.

Флаг дублирования (DUP) ДОЛЖЕН быть установлен на 1 Клиентом или Сервером, когда одна из сторон пытается повторно отправить пакет PUBLISH (ОПУБЛИКОВАТЬ) [MQTT-6.3.1.-1]. Флаг DUP ДОЛЖЕН быть установлен на 0 для всех сообщений с уровнем качества обслуживания (QoS) 0 [MQTT-6.3.1-2].

Значение флага дублирования (DUP) из принятого Сервером пакета PUBLISH не должно влиять на флаги пакета PUBLISH, отправляемого Сервером своим подписчикам. Флаг DUP в исходящем пакете PUBLISH устанавливается независимо от входящего пакета PUBLISH, его значение ДОЛЖНО быть определено только исходя из того, передается ли пакет PUBLISH повторно [MQTT-6.3.1-3].

Примечания

1 Получение Клиентом или Сервером управляющего пакета, содержащего флаг дублирования (DUP), установленный на 1, не является признаком того, что Клиент или Сервер уже получали более раннюю копию этого пакета.

2 Важно отметить, что флаг DUP относится к самому управляющему пакету, а не к содержащемуся в нем сообщению приложения. При использовании уровня качества обслуживания 1 Клиент может получить пакет PUBLISH с флагом DUP, установленным на 0, который содержит повторение сообщения приложения, которое он получил ранее, но с другим идентификатором пакета. В пункте 5.3.1 содержится дополнительная информация об идентификаторах пакетов.

6.3.1.2 Уровень качества обслуживания (QoS)

Позиция: байт 1, бит 2-1.

Это поле указывает уровень подтверждения доставки сообщения приложения. Уровни качества обслуживания (QoS) приведены в таблице 22.

Таблица 22 — Определения QoS

Значение QoS	Бит 1	Бит 2	Описание
0	0	0	Не более одной доставки
1	0	1	По крайней мере одна доставка
2	1	0	Ровно одна доставка
—	1	1	Зарезервировано — запрещено использовать

Пакет PUBLISH НЕ ДОЛЖЕН иметь оба бита QoS, установленные на 1. Если Сервер или Клиент получает пакет PUBLISH, у которого оба бита QoS установлены на 1, он ДОЛЖЕН прервать сетевое соединение [MQTT-6.3.1-4].

6.3.1.3 Флаг сохранения

Позиция: байт 1, бит 0.

Если флаг сохранения установлен на 1, в пакете PUBLISH, отправленном Клиентом на Сервер, Сервер ДОЛЖЕН хранить сообщение приложения и значение его уровня качества обслуживания, чтобы оно могло быть доставлено будущим подписчикам, подписки которых соответствуют названию темы сообщения [MQTT-6.3.1-5]. Когда установлена новая подписка, последнее сохраненное сообщение, при наличии, по каждой совпадающей теме ДОЛЖНО быть отправлено подписчику [MQTT-6.3.1-6]. Если Сервер получает сообщение с уровнем качества обслуживания 0 с флагом сохранения, установленным на 1, он ДОЛЖЕН удалить любое сообщение, ранее сохраненное для этой темы. СЛЕДУЕТ сохранить новое сообщение QoS 0 в качестве нового сохраненного сообщения для этой темы, но МОЖНО отказаться от него в любое время — если это произойдет, для этой темы не будет сохраненного сообщения [MQTT-6.3.1-7]. Дополнительную информацию о состоянии сохранения см. в подразделе 4.1.

При отправке пакета PUBLISH Клиенту Сервер ДОЛЖЕН установить флаг сохранения на 1, если сообщение отправлено в результате новой подписки, сделанной Клиентом [MQTT-6.3.1-8]. Он ДОЛЖЕН

установить флаг сохранения на 0, когда пакет PUBLISH отправляется Клиенту, поскольку он соответствует установленной подписке, независимо от того, как был установлен флаг в полученном сообщении [MQTT-6.3.1-9].

Пакет PUBLISH с флагом сохранения, установленным на 1, и полезная нагрузка, содержащая нулевые байты, будет обрабатываться как обычно Сервером и отправляться Клиентам с подпиской, соответствующей названию темы. Кроме того, любое существующее сохраненное сообщение с тем же именем темы ДОЛЖНО быть удалено, и любые будущие подписчики для темы не получат сохраненное сообщение [MQTT-6.3.1-10]. «Как обычно» означает, что флаг сохранения не установлен в сообщении, полученном существующими Клиентами. Сохраненное сообщение с нулевым байтом НЕ ДОЛЖНО храниться как сохраненное сообщение на Сервере [MQTT-6.3.1-11].

Если флаг сохранения установлен на 0, в пакете PUBLISH, отправленном Клиентом на Сервер, Сервер НЕ ДОЛЖЕН хранить это сообщение и НЕ ДОЛЖЕН удалять или заменять существующее сохраненное сообщение [MQTT-6.3.1-12].

Примечание — Сохраненные сообщения полезны, когда авторы отправляют сообщения о состоянии на нерегулярной основе. Новый абонент получит самую последнюю версию.

Поле «Остаток байтов»

Это размер переменного заголовка плюс размер полезной нагрузки.

6.3.2 Переменный заголовок

Переменный заголовок содержит следующие поля по порядку: название темы, идентификатор пакета.

6.3.2.1 Название темы

Название темы определяет информационный канал, в котором публикуются данные полезной нагрузки.

Название темы ДОЛЖНО присутствовать в качестве первого поля в заголовке пакета PUBLISH. Это ДОЛЖНА быть кодированная строка UTF-8 [MQTT-6.3.2-1], как определено в пункте 4.5.3.

Название темы в пакете PUBLISH НЕ ДОЛЖНО содержать подстановочных знаков [MQTT-6.3.2-2].

Название темы в пакете PUBLISH, отправленном Сервером Клиенту-подписчику, должно соответствовать фильтру темы подписки в соответствии с процессом сопоставления, определенным в Разделе 4.7 [MQTT-6.3.2-3]. Однако, поскольку Серверу разрешено заново определять название темы, оно может не совпадать с названием темы в исходном пакете PUBLISH.

6.3.2.2 Идентификатор пакета

Поле Идентификатор пакета присутствует только в пакетах PUBLISH, где уровень QoS равен 1 или 2. Пункт 5.3.1 предоставляет дополнительную информацию об идентификаторах пакетов.

6.3.2.3 Ненормативный пример переменного заголовка

В таблице 23 приведен примерный Переменный заголовок для пакета PUBLISH.

Таблица 23 — Ненормативный пример переменного заголовка пакета PUBLISH

Название темы	Описание	7	6	5	4	3	2	1	0
Байт 1	Длина MSB (0)	0	0	0	0	0	0	0	0
Байт 2	Длина LSB (3)	0	0	0	0	0	0	1	1
Байт 3	‘a’ (0x61)	0	1	1	0	0	0	0	1
Байт 4	‘/’ (0x2F)	0	0	1	0	1	1	1	1
Байт 5	‘b’ (0x62)	0	1	1	0	0	0	1	0
Идентификатор пакета									
Байт 6	Идентификатор пакета MSB (0)	0	0	0	0	0	0	0	0
Байт 7	Идентификатор пакета LSB (10)	0	0	0	0	1	0	1	0

Краткое описание ненормативного примера переменного заголовка пакета PUBLISH приведено в таблице 24.

Таблица 24 — Ненормативный пример пакета PUBLISH

Поле	Значение
Название темы	н/а
Идентификатор пакета	10

6.3.3 Полезная нагрузка

Полезная нагрузка содержит публикуемое сообщение приложения. Контент и формат данных специфичны для приложения. Длина полезной нагрузки может быть вычислена путем вычитания размера переменного заголовка из поля «Остаток байтов», которое находится в Фиксированном заголовке. Это действительно в том числе и для пакета PUBLISH, содержащего полезную нагрузку нулевого размера.

6.3.4 Ответ

Получатель пакета PUBLISH ДОЛЖЕН ответить согласно таблице 25, как определено уровнем качества обслуживания в пакете PUBLISH [MQTT-6.3.4-1].

Таблица 25 — Ожидаемый ответ пакета PUBLISH

Уровень QoS	Ожидаемый ответ
QoS 0	Нет
QoS 1	Пакет PUBACK
QoS 2	Пакет PUBREC

6.3.5 Действия

Клиент использует пакет PUBLISH для отправки сообщения приложения на Сервер для распространения среди Клиентов с соответствующими подписками.

Сервер использует пакет PUBLISH для отправки сообщения приложения каждому Клиенту, который имеет соответствующую подписку.

Когда Клиенты выполняют Подписки с фильтрами по темам, которые включают подстановочные знаки, становится возможным перекрытие фильтров, и опубликованное сообщение может соответствовать нескольким фильтрам. В этом случае Сервер ДОЛЖЕН доставлять сообщение Клиенту в соответствии с максимальным уровнем качества обслуживания всех соответствующих подписок [MQTT-6.3.5-1]. Кроме того, Сервер МОЖЕТ доставлять дополнительные копии сообщения, по одному для каждой соответствующей подписки, для соблюдения установленного уровня качества обслуживания сообщений в каждом случае.

Действия получателя после приема пакета PUBLISH, зависят от уровня QoS, как описано в подразделе 4.3.

Если Сервер в виде конкретной реализации, по некоторым причинам считает недопустимым получение пакета PUBLISH от Клиента, он не может сообщить об этом Клиенту. Сервер ДОЛЖЕН или подтвердить получение сообщения в соответствии с заданным уровнем качества обслуживания или прервать сетевое соединение [MQTT-6.3.5-2].

6.4 PUBACK — Подтверждение публикации

Пакет PUBACK — это ответ на пакет PUBLISH с уровнем QoS 1.

6.4.1 Фиксированный заголовок

Фиксированный заголовок пакета PUBACK приведен в таблице 26.

Таблица 26 — Фиксированный заголовок пакета PUBACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (4)				Зарезервировано			
	0	1	0	0	0	0	0	0
Байт 2	Остаток байтов (2)							
	0	0	0	0	0	0	1	0

Поле «Остаток байтов»

Это размер переменного заголовка. Для пакета PUBACK она имеет значение 2.

6.4.2 Переменный заголовок

Переменный заголовок пакета PUBACK, содержащий идентификатор пакета для пакета PUBLISH, приведен в таблице 27.

Таблица 27 — Переменный заголовок пакета PUBACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

6.4.3 Полезная нагрузка

Пакет PUBACK не имеет полезной нагрузки.

6.4.4 Действия

Описание приведено в пункте 7.3.2.

6.5 PUBREC — Подтверждение получения пакета PUBLISH с QoS 2

Пакет PUBREC — это ответ на пакет PUBLISH с уровнем качества обслуживания (QoS) 2. Это второй пакет при организации обмена по протоколу с уровнем качества обслуживания 2.

6.5.1 Фиксированный заголовок

Фиксированный заголовок пакета PUBACK приведен в таблице 28.

Таблица 28 — Фиксированный заголовок пакета PUBACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (1)				Зарезервировано			
	0	1	0	1	0	0	0	0
Байт 2	Остаток байтов (2)							
	0	0	0	0	0	0	1	0

Поле «Остаток байтов»

Это размер переменного заголовка. Для пакета PUBREC она имеет значение 2.

6.5.2 Переменный заголовок

Переменный заголовок содержит идентификатор пакета из пакета PUBLISH, который подтверждается. Переменный заголовок пакета PUBREC приведен в таблице 29.

Таблица 29 — Переменный заголовок пакета PUBREC

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

6.5.3 Полезная нагрузка

Пакет PUBREC не имеет полезной нагрузки.

6.5.4 Действия

Полностью описано в пункте 7.3.3.

6.6 PUBREL — Выпуск пакета PUBLISH с QoS 2

Пакет PUBREL является ответом на пакет PUBREC. Это третий пакет при организации обмена по протоколу с уровнем качества обслуживания (QoS) 2.

6.6.1 Фиксированный заголовок

Фиксированный заголовок пакета PUBREL приведен в таблице 30.

Таблица 30 — Фиксированный заголовок пакета PUBREL

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (6)				Зарезервировано			
	0	1	1	0	0	0	1	0
Байт 2	Остаток байтов (2)							
	0	0	0	0	0	0	1	0

Биты 3, 2, 1 и 0 фиксированного заголовка в пакете управления PUBREL зарезервированы и ДОЛЖНЫ быть установлены на 0, 0, 1 и 0 соответственно. Сервер ДОЛЖЕН считать любое другое значение искаженным и прерывать сетевое соединение [MQTT-6.6.1-1].

Поле «Остаток байтов»

Это размер переменного заголовка. Для пакета PUBREL это значение равно 2.

6.6.2 Переменный заголовок

Переменный заголовок содержит тот же идентификатор пакета, что и пакет PUBREC, который подтверждается. Переменный заголовок пакета PUBREL приведен в таблице 31.

Таблица 31 — Переменный заголовок пакета PUBREL

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

6.6.3 Полезная нагрузка

Пакет PUBREL не имеет полезной нагрузки.

6.6.4 Действия

Описание приведено в пункте 7.3.3.

6.7 PUBCOMP — Публикация завершена

Пакет PUBCOMP — это ответ на пакет PUBREL. Это четвертый и последний пакет при организации обмена по протоколу с уровнем качества обслуживания 2.

6.7.1 Фиксированный заголовок

Фиксированный заголовок пакета PUBCOMP приведен в таблице 32.

Таблица 32 — Фиксированный заголовок пакета PUBCOMP

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (7)				Зарезервировано			
	0	1	1	1	0	0	0	0
Байт 2	Остаток байтов (2)							
	0	0	0	0	0	0	1	0

Поле «Остаток байтов»

Это размер переменного заголовка. Для пакета PUBCOMP это значение равно 2.

6.7.2 Переменный заголовок

Переменный заголовок содержит тот же идентификатор пакета, что и пакет PUBREL, который подтверждается. Переменный заголовок пакета PUBCOMP приведен в таблице 33.

Таблица 33 — Переменный заголовок пакета PUBCOMP

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

6.7.3 Полезная нагрузка

Пакет PUBCOMP не имеет полезной нагрузки.

6.7.4 Действия

Описание приведено в пункте 7.3.3.

6.8 SUBSCRIBE — Подписка на темы

Пакет SUBSCRIBE отправляется Клиентом на Сервер для создания одной или нескольких Подписок. Каждая подписка подтверждает интерес Клиента к одной или нескольким темам. Сервер отправляет пакеты PUBLISH Клиенту, чтобы перенаправить сообщения приложений, опубликованные в темах, соответствующих подпискам Клиента. Пакет SUBSCRIBE также определяет (для каждой подписки) максимальный уровень качества обслуживания (QoS), который Сервер должен обеспечивать при отправке сообщений приложения Клиенту.

6.8.1 Фиксированный заголовок

Фиксированный заголовок пакета SUBSCRIBE приведен в таблице 34.

Таблица 34 — Фиксированный заголовок пакета SUBSCRIBE

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (8)				Зарезервировано			
	1	0	0	0	0	0	1	0
Байт 2	Остаток байтов (2)							

Биты 3, 2, 1 и 0 фиксированного заголовка управляющего пакета SUBSCRIBE зарезервированы и ДОЛЖНЫ быть установлены на 0, 0, 1 и 0 соответственно. Сервер ДОЛЖЕН обработать любое другое значение как искаженное и прервать сетевое соединение [MQTT-6.8.1-1].

Поле «Остаток байтов»

Это размер переменного заголовка (2 байта) плюс длина полезной нагрузки.

6.8.2 Переменный заголовок

Переменный заголовок содержит идентификатор пакета. В пункте 5.3.1 содержится дополнительная информация об идентификаторах пакетов.

Пример — В таблице 35 приведен переменный заголовок с идентификатором пакета, установленным на 10.

Таблица 35 — Переменный заголовок с идентификатором пакета 10

	Описание	7	6	5	4	3	2	1	0
Идентификатор пакета									
Байт 1	Идентификатор пакета MSB (0)	0	0	0	0	0	0	0	0
Байт 2	Идентификатор пакета LSB (10)	0	0	0	0	1	0	1	0

6.8.3 Полезная нагрузка

Полезная нагрузка пакета SUBSCRIBE содержит список фильтров по темам, в котором указаны темы, на которые Клиент намерен подписаться. Фильтры тем в полезной нагрузке пакета SUBSCRIBE ДОЛЖНЫ быть закодированными UTF-8 строками, как определено в пункте 4.5.3 [MQTT-6.8.3-1]. Сервер ДОЛЖЕН поддерживать фильтры тем, содержащие символы подстановки, определенные в пункте 7.7.1. Если реализация Сервера не поддерживает фильтры тем, которые содержат подстановочные знаки, он ДОЛЖЕН отклонить любой запрос на подписку, в котором содержатся подобные фильтры [MQTT-6.8.3-2]. За каждым фильтром следует байт, называемый запрошенным QoS. Он определяет

максимальный уровень качества обслуживания, который должен обеспечиваться Сервером при отправке сообщений приложений Клиенту.

Полезная нагрузка пакета SUBSCRIBE ДОЛЖНА содержать по меньшей мере одно сочетание фильтра темы и запрошенного QoS. Пакет SUBSCRIBE без полезной нагрузки является нарушением протокола [MQTT-6.8.3-3]. Информация об обработке ошибок приведена в подразделе 7.8.

Поле запрошенного максимального уровня качества обслуживания закодировано в байте, следующем за каждым именем темы в кодировке UTF-8, и эти сочетания фильтров тем и запрошенных QoS формируют непрерывный пакет.

Формат полезной нагрузки пакета SUBSCRIBE приведен в таблице 36.

Таблица 36 — Формат полезной нагрузки пакета SUBSCRIBE

Описание	7	6	5	4	3	2	1	0
Фильтр тем								
Байт 1	Длина MSB							
Байт 2	Длина LSB							
Байт 3 ... N	Фильтр тем							
Запрошенный QoS								
	Зарезервировано						QoS	
Байт N+1	0	0	0	0	0	0	X	X

Верхние 6 бит байта запрошенного QoS не используются в текущей версии протокола. Они зарезервированы для будущего использования. Сервер ДОЛЖЕН считать пакет SUBSCRIBE искаженным и прерывать сетевое соединение, если любой из зарезервированных битов в полезной нагрузке отличен от нуля, или QoS не равно 0, 1 или 2 [MQTT-6-8.3-4].

Пример — В таблице 37 приведена полезная нагрузка для пакета SUBSCRIBE.

Таблица 37 — Байтовый формат полезной нагрузки для пакета SUBSCRIBE

	Описание	7	6	5	4	3	2	1	0
Фильтр по темам									
Байт 1	Длина MSB (0)	0	0	0	0	0	0	0	0
Байт 2	Длина LSB (3)	0	0	0	0	0	0	1	1
Байт 3	‘a’ (0x61)	0	1	1	0	0	0	0	1
Байт 4	‘/’ (0x2F)	0	0	1	0	1	1	1	1
Байт 5	‘b’ (0x62)	0	1	1	0	0	0	1	0
Запрошенный QoS									
Байт 6	Запрошенный QoS (1)	0	0	0	0	0	0	0	1
Фильтр по темам									
Байт 7	Длина MSB (0)	0	0	0	0	0	0	0	0
Байт 8	Длина LSB (3)	0	0	0	0	0	0	1	1
Байт 9	‘c’ (0x63)	0	1	1	0	0	0	1	1
Байт 10	‘/’ (0x2F)	0	0	1	0	1	1	1	1
Байт 11	‘d’ (0x64)	0	1	1	0	0	1	0	0
Запрошенный QoS									
Байт 12	Запрошенный QoS (2)	0	0	0	0	0	0	1	0

Описание полезной нагрузки для пакета SUBSCRIBE приведено в таблице 38.

Таблица 38 — Пример полезной нагрузки

Название темы	“a/b”
Запрошенный QoS	0x01
Название темы	“c/d”
Запрошенный QoS	0x02

6.8.4 Ответ

При получении Сервером пакета SUBSCRIBE от Клиента, Сервер ДОЛЖЕН отвечать с пакетом SUBACK [MQTT-6.8.4-1]. Пакет SUBACK ДОЛЖЕН иметь тот же идентификатор пакета, что и пакет SUBSCRIBE, который он подтверждает [MQTT-6.8.4-2].

Серверу разрешено начинать отправку пакетов PUBLISH, соответствующих Подписке, до того, как Сервер отправит пакет SUBACK.

Если Сервер получает пакет SUBSCRIBE, содержащий фильтр тем, который идентичен существующему фильтру тем Подписки, он ДОЛЖЕН полностью заменить существующую Подписку новой Подпиской. Фильтр темы в новой Подписке будет идентичен фильтру темы в предыдущей Подписке, хотя его максимальное значение уровня качества обслуживания (QoS) может отличаться. Любые существующие сохраненные сообщения, соответствующие фильтру темы, ДОЛЖНЫ быть повторно отправлены, но поток публикаций НЕ ДОЛЖЕН прерываться [MQTT-6.8.4-3].

Если фильтр темы не идентичен существующему фильтру Подписки, создается новая Подписка и отправляются все соответствующие сохраненные сообщения.

Если Сервер получает пакет SUBSCRIBE, который содержит несколько фильтров тем, он ДОЛЖЕН обрабатывать этот пакет, как если бы он получил последовательность из нескольких пакетов SUBSCRIBE, за исключением того, что он объединяет ответы на них в один ответ SUBACK [MQTT-6.8.4-4].

Пакет SUBACK, отправленный Сервером Клиенту, ДОЛЖЕН содержать код возврата для каждого сочетания фильтров тем и запрошенных QoS. Этот код возврата ДОЛЖЕН или обозначить максимальный уровень качества обслуживания, который был обеспечен Сервером для этой Подписки, или указать на сбой подписки [MQTT-6.8.4-5]. Сервер может обеспечить более низкий уровень качества обслуживания, чем запросил подписчик. Для пересылаемых сообщений приложения Сервер ДОЛЖЕН обеспечить такой же уровень качества обслуживания, что был указан в полученных сообщениях. Если это невозможно, то Сервер ДОЛЖЕН обеспечить максимально возможный уровень качества обслуживания, предусмотренный реализацией. Серверу разрешено отправлять дубликаты сообщений подписчику в случае, когда исходное сообщение было опубликовано с QoS 1, а максимальным предоставленным QoS был QoS 0 [MQTT-6.8.4-6].

Примеры

1 Если Клиенту-подписчику был предоставлен максимальный QoS 1 для конкретного фильтра тем, то сообщение приложения QoS 0, соответствующее фильтру, доставляется Клиенту при QoS 0. Это означает, что Клиент получает не более одной копии сообщения. С другой стороны, сообщение QoS 2, опубликованное по той же теме, было понижено Сервером до QoS 1 для доставки Клиенту, чтобы Клиент мог получать дубликаты копий сообщения.

2 Если Клиенту-подписчику предоставлен максимальный QoS 0, тогда сообщение приложения, первоначально опубликованное как QoS 2, может потеряться при переходе к Клиенту, но Сервер не должен отправлять дубликат этого сообщения. Сообщение QoS 1, опубликованное в той же теме, может быть потеряно или дублировано при его передаче такому Клиенту.

Примечание — Подписка на фильтр тем с обеспеченным уровнем качества обслуживания 2 эквивалентна высказыванию «Я хотел бы получать сообщения, соответствующие этому фильтру с тем уровнем качества обслуживания, с каким они были опубликованы». Это означает, что издатель несет ответственность за определение максимального уровня качества обслуживания, который будет обеспечен при доставке сообщения, но подписчик может потребовать от Сервера понизить этот уровень при необходимости.

6.9 SUBACK — Подтверждение подписки

Пакет SUBACK отправляется Сервером Клиенту для подтверждения получения и обработки пакета SUBSCRIBE.

Пакет SUBACK содержит список кодов возврата, которые определяют максимальный уровень QoS, который был предоставлен по каждой Подписке, запрошенной пакетом SUBSCRIBE.

6.9.1 Фиксированный заголовок

Фиксированный заголовок пакета SUBACK приведен в таблице 39.

Таблица 39 — Фиксированный заголовок пакета SUBACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (9)				Зарезервировано			
	1	0	0	1	0	0	0	0
Байт 2	Остаток байтов							

Поле «Остаток байтов»

Это размер переменного заголовка (2 байта) плюс длина полезной нагрузки.

6.9.2 Переменный заголовок

Переменный заголовок содержит идентификатор пакета из пакета SUBSCRIBE, который подтверждается. В таблице 40 приведен формат переменного заголовка.

Таблица 40 — Переменный заголовок пакета SUBACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

6.9.3 Полезная нагрузка

Полезная нагрузка содержит список кодов возврата. Каждый код возврата соответствует фильтру тем в подтвержденном пакете SUBSCRIBE. Порядок кодов возврата в пакете SUBACK ДОЛЖЕН соответствовать порядку фильтров тем в пакете SUBSCRIBE [MQTT-6.9.3-1].

В таблице 41 приведено поле кода возврата, закодированное в байте в полезную нагрузку.

Таблица 41 — Формат полезной нагрузки пакета SUBACK

Бит	7	6	5	4	3	2	1	0
	Код возврата							
Байт 1	X	0	0	0	0	0	X	X

Разрешенные коды возврата:

0x00 — Успешно — Макс. QoS 0

0x01 — Успешно — Макс. QoS 1

0x02 — Успешно — Макс. QoS 2

0x80 — Отказ

Коды возврата SUBACK, отличные от 0x00, 0x01, 0x02 и 0x80, зарезервированы и НЕ ДОЛЖНЫ использоваться [MQTT-6.9.3-2].

Пример — В таблице 42 приведен байтовый формат полезной нагрузки для пакета SUBACK.

Таблица 42 — Пример полезной нагрузки

	Описание	7	6	5	4	3	2	1	0
Байт 1	Успешно — Максимальное QoS 0	0	0	0	0	0	0	0	0
Байт 2	Успешно — Максимальное QoS 2	0	0	0	0	0	0	1	0
Байт 3	Отказ	1	0	0	0	0	0	0	0

Краткое описание пакета SUBACK приведено в таблице 43.

Таблица 43 — Пример полезной нагрузки

Наименование	Значение
Успешно — Максимальное QoS 0	0
Успешно — Максимальное QoS 2	2
Отказ	128

6.10 UNSUBSCRIBE — Отменить подписку на темы

Пакет UNSUBSCRIBE отправляется Клиентом на Сервер с целью отказа от подписки на темы.

6.10.1 Фиксированный заголовок

Фиксированный заголовок пакета UNSUBSCRIBE приведен в таблице 44.

Таблица 44 — Фиксированный заголовок пакета UNSUBSCRIBE

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (10)				Зарезервировано			
	1	0	1	0	0	0	1	0
Байт 2	Остаток байтов							

Биты 3, 2, 1 и 0 фиксированного заголовка управляющего пакета UNSUBSCRIBE зарезервированы и ДОЛЖНЫ быть установлены на 0, 0, 1 и 0 соответственно. Сервер ДОЛЖЕН считать любое другое значение искаженным и прервать сетевое соединение [MQTT-6.10.1-1].

Поле «Остаток байтов»

Это размер переменного заголовка (2 байта) плюс длина полезной нагрузки.

6.10.2 Переменный заголовок

Переменный заголовок содержит идентификатор пакета. В пункте 5.3.1 содержится дополнительная информация об идентификаторах пакетов. В таблице 45 приведен переменный заголовок пакета UNSUBSCRIBE.

Таблица 45 — Переменный заголовок пакета UNSUBSCRIBE

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

6.10.3 Полезная нагрузка

Полезная нагрузка для пакета UNSUBSCRIBE содержит список фильтров тем, от которых Клиент намерен отписаться. Фильтры тем в пакете UNSUBSCRIBE ДОЛЖНЫ быть закодированными строками UTF-8, как определено в пункте 4.5.3, формировать непрерывный пакет данных [MQTT-6.10.3-1].

Полезная нагрузка пакета UNSUBSCRIBE ДОЛЖНА содержать, как минимум, один фильтр тем. Пакет UNSUBSCRIBE без полезной нагрузки является нарушением протокола [MQTT-6.10.3-2]. Информация по обработке ошибок приведена в подразделе 7.8.

Пример — В таблице 46 приведен пример полезной нагрузки для пакета UNSUBSCRIBE.

Таблица 46 — Пример байтового формата полезной нагрузки

	Описание	7	6	5	4	3	2	1	0
Фильтр по темам									
Байт 1	Длина MSB (0)	0	0	0	0	0	0	0	0
Байт 2	Длина LSB (3)	0	0	0	0	0	0	1	1
Байт 3	‘a’ (0x61)	0	1	1	0	0	0	0	1

Окончание таблицы 46

	Описание	7	6	5	4	3	2	1	0
Байт 4	'/' (0x2F)	0	0	1	0	1	1	1	1
Байт 5	'b' (0x62)	0	1	1	0	0	0	1	0
Фильтр по темам									
Байт 6	Длина MSB (0)	0	0	0	0	0	0	0	0
Байт 7	Длина LSB (3)	0	0	0	0	0	0	1	1
Байт 8	'c' (0x63)	0	1	1	0	0	0	1	1
Байт 9	'/' (0x2F)	0	0	1	0	1	1	1	1
Байт 10	'd' (0x64)	0	1	1	0	0	1	0	0

Пакет UNSUSCRIBE кратко описан в таблице 47.

Таблица 47 — Пример полезной нагрузки

Наименование	Тема
Название темы	"a/b"
Название темы	"c/d"

6.10.4 Ответ

Фильтры тем (независимо от того, содержат ли они подстановочные знаки или нет), указанные в пакете UNSUBSCRIBE, ДОЛЖНЫ сравниваться по каждому символу с текущим набором фильтров тем, хранящихся на Сервере для Клиента. Если найдено точное соответствие, то подписка удаляется, в противном случае никакой дополнительной обработки не происходит [MQTT-6.10.4-1].

Если Сервер удаляет подписку:

- он ДОЛЖЕН прекратить добавлять новые сообщения в очередь на пересылку [MQTT-6.10.4-2];
- он ДОЛЖЕН завершить доставку любых сообщений QoS 1 или QoS 2, которые он начал отправлять Клиенту [MQTT-6.10.4-3];
- он МОЖЕТ продолжить доставку любых существующих сообщений, сохраненных в буфере для доставки Клиенту.

Сервер ДОЛЖЕН отвечать на запрос UNSUBCRIBE, отправив пакет UNSUBACK. Пакет UNSUBACK ДОЛЖЕН иметь тот же идентификатор пакета, что и пакет UNSUBSCRIBE [MQTT-6.10.4-4]. Даже если никакие Подписки на темы не удалены, Сервер ДОЛЖЕН отправить UNSUBACK [MQTT-6.10.4-5].

Если Сервер получает пакет UNSUBSCRIBE, содержащий несколько фильтров тем, он ДОЛЖЕН обрабатывать этот пакет, как если бы он получил последовательность из нескольких пакетов UNSUBSCRIBE, за исключением того, что он отправляет только один ответ UNSUBACK [MQTT-6.10.4-6].

6.11 UNSUBACK — Подтверждение отмены подписки

Пакет UNSUBACK отправляется Сервером Клиенту для подтверждения получения пакета UNSUBSCRIBE.

6.11.1 Фиксированный заголовок

Фиксированный заголовок пакета UNSUBACK приведен в таблице 48.

Таблица 48 — Фиксированный заголовок пакета UNSUBACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (11)				Зарезервировано			
	1	0	1	1	0	0	0	0
Байт 2	Остаток байтов (2)							
	0	0	0	0	0	0	1	0

Поле «Остаток байтов»

Это размер переменного заголовка. Для пакета UNSUBACK данное значение равно 2.

6.11.2 Переменный заголовок

Переменный заголовок содержит идентификатор пакета UNSUBSCRIBE, который подтверждается. Переменный заголовок пакета UNSUBACK приведен в таблице 49.

Таблица 49 — Переменный заголовок пакета UNSUBACK

Бит	7	6	5	4	3	2	1	0
Байт 1	Идентификатор пакета MSB							
Байт 2	Идентификатор пакета LSB							

6.11.3 Полезная нагрузка

Пакет UNSUBACK не имеет полезной нагрузки.

6.12 PINGREQ — Запрос PING

Пакет PINGREQ отправляется Клиентом на Сервер. Его можно использовать для того, чтобы:

- указать Серверу, что Клиент активен, при отсутствии каких-либо других управляющих пакетов, отправляемых Клиентом на Сервер;
- направить запрос для получения подтверждения, что Сервер активен;
- провести тест сети передачи данных, чтобы подтвердить, что сетевое подключение активно.

Данный пакет используется для обработки активного соединения, более подробная информация приведена в подпункте 6.1.2.10.

6.12.1 Фиксированный заголовок

Фиксированный заголовок пакета PINGREQ приведен в таблице 50.

Таблица 50 — Фиксированный заголовок пакета PINGREQ

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (12)				Зарезервировано			
	1	1	0	0	0	0	0	0
Байт 2	Остаток байтов (0)							
	0	0	0	0	0	0	0	0

6.12.2 Переменный заголовок

Пакет PINGREQ не имеет переменного заголовка.

6.12.3 Полезная нагрузка

Пакет PINGREQ не имеет полезной нагрузки.

6.12.4 Ответ

Сервер ДОЛЖЕН отправить пакет PINGRESP в ответ на пакет PINGREQ [MQTT-6.12.4-1].

6.13 PINGRESP — Ответ PING

Пакет PINGRESP отправляется Сервером Клиенту в ответ на пакет PINGREQ. Он указывает, что Сервер активен.

Данный пакет используется для поддержки активного соединения между Сервером и Клиентом, более подробная информация приведена в подпункте 6.1.2.10.

6.13.1 Фиксированный заголовок

Фиксированный заголовок пакета PINGRESP приведен в таблице 51.

Таблица 51 — Фиксированный заголовок пакета PINGRESP

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (13)				Зарезервировано			
	1	1	0	1	0	0	0	0
Байт 2	Остаток байтов (0)							
	0	0	0	0	0	0	0	0

6.13.2 Переменный заголовок

Пакет PINGRESP не имеет переменного заголовка.

6.13.3 Полезная нагрузка

Пакет PINGRESP не имеет полезной нагрузки.

6.14 DISCONNECT — Уведомление об отключении

Пакет DISCONNECT является последним управляющим пакетом, отправляемым Клиентом на Сервер. Он указывает, что Клиент отключается преднамеренно, в соответствии с процедурой, установленной настоящим стандартом.

6.14.1 Фиксированный заголовок

Фиксированный заголовок пакета DISCONNECT приведен в таблице 52.

Таблица 52 — Фиксированный заголовок пакета DISCONNECT

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип управляющего пакета MQTT (14)				Зарезервировано			
	1	1	1	0	0	0	0	0
Байт 2	Остаток байтов (0)							
	0	0	0	0	0	0	0	0

Сервер ДОЛЖЕН проверить, чтобы зарезервированные биты были установлены на ноль и отключить Клиента, если они не равны нулю [MQTT-6.14.1-1].

6.14.2 Переменный заголовок

Пакет DISCONNECT не имеет переменного заголовка.

6.14.3 Полезная нагрузка

Пакет DISCONNECT не имеет полезной нагрузки.

6.14.4 Ответ

После отправки пакета DISCONNECT Клиент:

ДОЛЖЕН разорвать сетевое соединение [MQTT-6.14.4-1];

больше НЕ ДОЛЖЕН отправлять управляющие пакеты через это сетевое подключение MQTT-6.14.4-2].

При получении DISCONNECT Сервер:

ДОЛЖЕН удалить любое сообщение «последней воли», связанное с текущим подключением, без его публикации, как описано в разделе 6.1.2.5 [MQTT-6.14.4-3];

СЛЕДУЕТ прервать сетевое соединение, если Клиент еще этого не сделал.

7 Описание работы протоколов

7.1 Сохранение состояния

Клиенту и Серверу необходимо сохранять состояние Сеанса, чтобы должным образом обеспечивать необходимый уровень качества обслуживания (QoS). Клиент и Сервер ДОЛЖНЫ сохранять состояние Сеанса на протяжении всего Сеанса [MQTT-7.1.0-1]. Сеанс ДОЛЖЕН длиться как минимум до тех пор, пока активно сетевое соединение [MQTT-7.1.0-2].

Сохраненные сообщения не являются частью сохраненного состояния Сеанса на Сервере. Серверу СЛЕДУЕТ сохранять такие сообщения, пока их не удалит Клиент.

Примечания

1 Объем хранилищ Клиента и Сервера, конечно, ограничен. Сами хранилища могут быть объектами политик администрирования, например, в части максимального времени хранения состояния Сеанса между повторным установлением сетевого соединения. Сохраненное состояние Сеанса может быть аннулировано в результате действий администратора, включая автоматический ответ на определенные условия. Это приводит к завершению Сеанса. Данные действия могут быть вызваны ограничениями ресурсов или другими операционными причинами. Будет разумным оценить возможности хранилищ Клиента и Сервера, чтобы гарантировать их достаточность.

2 Сбои в аппаратном или программном обеспечении могут привести к потере или повреждению состояния Сеанса, сохраненного Клиентом или Сервером.

3 При нормальном функционировании Клиента или Сервера причиной потери или повреждения сохраненного состояния Сеанса могут стать действия администратора, сбой программного или аппаратного обеспечения. Действия администратора могут быть автоматическим ответом на определенные условия. Данные действия могут быть вызваны ограничениями ресурсов или другими операционными причинами. Например, Сервер может определить, что, согласно внешней информации, сообщение или сообщения больше не могут доставляться текущему или будущему Клиенту.

4 Пользователи протокола MQTT должны оценивать возможности хранилищ тех или иных реализаций Клиента и Сервера, для должного обеспечения их собственных нужд.

Примеры

1 Например, пользователь, желающий собрать показания счетчиков электроэнергии, может решить, что ему необходимо использовать сообщения QoS 1, поскольку он должен защитить показания от потери по сети, однако он может определить, что источник питания достаточно надежный, чтобы данные у Клиента и Сервера могли храниться в энергозависимой памяти без особого риска потери.

2 И, наоборот, создатель приложения по оплате парковки может решить, что платежное сообщение не может быть потеряно ни при каких обстоятельствах, поэтому требуется, чтобы все данные были записаны в энергонезависимой памяти до того, как они будут переданы по сети.

7.2 Сетевые подключения

Для работы протокола MQTT требуется базовый протокол передачи данных, который обеспечит упорядоченную передачу потока байтов без потерь от Клиента к Серверу и от Сервера к Клиенту.

Примечания

1 Транспортным протоколом, используемым для работы MQTT 3.1, был TCP/IP, как определено в [6]. TCP/IP может использоваться для MQTT 3.1.1. Также подходят следующие:

- TLS [3];
- WebSocket [4].

2 TCP-порты 8883 и 1883 зарегистрированы в IANA в отношении MQTT TLS и отличной от TLS коммуникации, соответственно.

Протоколы передачи данных, не предусматривающие предварительного установления соединения, такие как UDP, не подходят для обеспечения работы MQTT, поскольку они могут потерять или изменить порядок данных.

7.3 Уровни качества обслуживания и потоки протоколов

MQTT обеспечивает доставку сообщений приложений в соответствии с уровнями качества обслуживания (QoS), определенными в настоящем стандарте. Протокол доставки является симметричным, в описании ниже Клиент и Сервер могут выполнять роль Отправителя или Получателя. Протокол доставки описывает исключительно доставку сообщения приложения от одного отправителя к одному получателю. Когда Сервер доставляет сообщение приложения нескольким Клиентам, каждый Клиент обрабатывается независимо. Уровень качества обслуживания, обеспечиваемый при доставке исходящего сообщения приложения Клиенту, может отличаться от такового для входящего сообщения приложения.

Ненормативные блок-схемы в следующих разделах предназначены для демонстрации возможных подходов к реализации.

7.3.1 QoS 0: Доставка не более одного раза

Сообщение доставляется в соответствии с возможностями базовой сети. Получатель не отправляет ответ, и отправитель не выполняет никаких повторных попыток отправки. Сообщение доставляется получателю только один раз или не доставляется вовсе.

По протоколу доставки QoS 0 отправитель:

- ДОЛЖЕН отправить пакет PUBLISH с QoS = 0, DUP = 0 [MQTT-7.3.1-1].

По протоколу доставки QoS 0 получатель:

- принимает сообщение при получении пакета PUBLISH.

В таблице 53 в качестве примера приведена технологическая схема протокола QoS.

Таблица 53 — Технологическая схема протокола QoS 0

Действия отправителя	Управляющий пакет	Действия получателя
PUBLISH QoS 0, DUP=0		
	---------->	
		Отправить сообщение приложения соответствующему получателю (получателям)

7.3.2 QoS 1: Доставка не менее одного раза

При данном уровне качества обслуживания гарантируется, что сообщение будет доставлено получателю не менее одного раза. Пакет PUBLISH QoS 1 имеет идентификатор пакета в переменном заголовке и подтверждается пакетом PUBACK. Дополнительная информация об идентификаторах пакетов указана в пункте 5.3.1.

В протоколе доставки QoS 1 отправитель ДОЛЖЕН:

- назначать неиспользуемый идентификатор пакета каждый раз, когда у него есть новое сообщение приложения для публикации;
- отправить пакет PUBLISH, содержащий этот идентификатор пакета с QoS = 1, DUP = 0;
- рассматривать пакет PUBLISH как «неподтвержденный», пока он не получит соответствующий пакет PUBACK от получателя. Более подробная информация о неподтвержденных сообщениях приведена в подразделе 7.4.

[MQTT-7.3.2-1].

Идентификатор пакета становится доступным для повторного использования, как только отправитель получил соответствующий пакет PUBACK.

Обратите внимание, что отправителю разрешено отправлять дальнейшие пакеты PUBLISH с различными идентификаторами пакетов, пока он ожидает получения подтверждений.

В протоколе доставки QoS 1 получатель ДОЛЖЕН:

- приняв сообщение приложения из пакета PUBLISH, ответить пакетом PUBACK, содержащим соответствующий идентификатор пакета;
- после отправки пакета PUBACK считать любой входящий пакет PUBLISH, который содержит тот же идентификатор пакета, новой публикацией, независимо от установки флага DUP.

[MQTT-7.3.2-2].

Технологическая схема протокола уровня качества обслуживания (QoS) 1 приведена в таблице 54.

Таблица 54 — Технологическая схема протокола QoS 1

Действия отправителя	Управляющий пакет	Действия получателя
Сохранить сообщение		
Отправить PUBLISH QoS 1, DUP 0, <Идентификатор пакета>	--------->	
		Инициировать отправку сообщения приложения[1)]
	<---------	Отправить PUBACK <Идентификатор пакета>
Аннулировать сообщение		
[1)] Получателю не требуется доставлять сообщение приложения перед отправкой PUBACK. Когда его первоначальный отправитель получает пакет PUBACK, право собственности на сообщение приложения переходит к получателю.		

7.3.3 QoS 2: Ровно одна доставка

Данный уровень качества обслуживания является самым высоким уровнем качества обслуживания для использования в случаях, когда неприемлемы ни потери, ни дублирование сообщений. При передаче сообщений с данным уровнем качества обслуживания возрастают накладные расходы.

Сообщение QoS 2 имеет идентификатор пакета в переменном заголовке. В пункте 5.3.1 содержится дополнительная информация об идентификаторах пакетов. Получатель пакета PUBLISH с QoS 2 инициализирует процесс подтверждения, состоящий из двух этапов.

В протоколе доставки QoS 2 отправитель:

- ДОЛЖЕН назначить неиспользуемый идентификатор пакета, если у него есть новое сообщение приложения для публикации;
- ДОЛЖЕН отправить пакет PUBLISH, содержащий данный идентификатор пакета, с QoS = 2, DUP = 0;
- ДОЛЖЕН считать пакет PUBLISH «неподтвержденным», пока он не получит соответствующий пакет PUBREC от получателя. См. информацию о неподтвержденных сообщениях в подразделе 7.4;
- ДОЛЖЕН отправить пакет PUBREL, когда он получает пакет PUBREC от получателя. Данный пакет PUBREL ДОЛЖЕН содержать тот же идентификатор пакета, что и исходный пакет PUBLISH;
- ДОЛЖЕН считать пакет PUBREL «неподтвержденным», пока он не получит соответствующий пакет PUBCOMP от получателя;
- НЕ МОЖЕТ повторно отправить PUBLISH после отправки соответствующего пакета PUBREL [MQTT-7.3.3-1].

Идентификатор пакета становится доступным для повторного использования, как только отправитель получил соответствующий пакет PUBCOMP.

Обратите внимание, что отправителю разрешено отправлять следующие пакеты PUBLISH с различными идентификаторами пакетов, пока он ожидает получения подтверждений.

В протоколе доставки QoS 2 получатель ДОЛЖЕН:

- приняв сообщение приложения из пакета PUBLISH, ответить пакетом PUBREC, содержащим соответствующий идентификатор пакета;
- до тех пор, пока он не получит соответствующий пакет PUBREL, подтверждать любой последующий пакет PUBLISH с тем же идентификатором пакета, отправив PUBREC. В этом случае он НЕ ДОЛЖЕН инициировать дублирование сообщений для последующих получателей;
- отвечать на пакет PUBREL, отправив пакет PUBCOMP, содержащий тот же идентификатор пакета, что и PUBREL;
- после того, как он отправил PUBCOMP, считать любой последующий пакет PUBLISH, содержащий данный идентификатор пакета, новой публикацией [MQTT-7.3.3-2].

Ненормативный пример технологической схемы протокола уровня качества обслуживания (QoS) 2 приведен в таблице 55.

Таблица 55 — Технологическая схема протокола QoS 2

Действия отправителя	Управляющий пакет	Действия получателя
Сохранить сообщение		
PUBLISH QoS 2, DUP 0, <Идентификатор пакета>		
	-------->	
		Способ А, сохранить сообщение или Способ В, сохранить <Идентификатор пакета>, затем инициировать передачу доставки сообщения приложения[1)]
		PUBREC <Идентификатор пакета>
	<--------	

Окончание таблицы 55

Действия отправителя	Управляющий пакет	Действия получателя
Аннулировать сообщение, сохранить полученный PUBREC <Идентификатор пакета>		
PUBREL <Идентификатор пакета>		
	--------->	
		Способ A, инициировать отправку сообщения приложения 1, затем отменить сообщение; или Способ B, аннулировать <Идентификатор пакета>
		Отправить PUBCOMP <Идентификатор пакета>
	<---------	
Аннулировать сохраненное состояние		
[1)] Получателю не требуется доставлять сообщение приложения перед отправкой PUBREC или PUBCOMP. Когда его первоначальный отправитель получает пакет PUBREC, право собственности на сообщение приложения переходит к получателю.		

В таблице 55 описаны два способа, с помощью которых QoS 2 может обрабатываться получателем. Они различаются в той точке, в которой сообщение становится доступным для последующей доставки. Выбор способа A или способа B является конкретным для каждого случая. Если при доставке выбран один из этих подходов, это не влияет на гарантии потока QoS 2.

7.4 Повторная доставка сообщений

Когда Клиент повторно подключается с флагом очистки Сеанса, установленному на 0, Клиент и Сервер ДОЛЖНЫ повторно отправлять любые неподтвержденные пакеты PUBLISH (где QoS> 0) и пакеты PUBREL с использованием их исходных идентификаторов пакетов [MQTT-7.4.0-1]. Это единственное обстоятельство, когда Клиент или Сервер ДОЛЖНЫ повторно отправлять сообщения.

Примечание — Исторически повторная передача управляющих пакетов требовалась для преодоления потери данных в некоторых старых сетях TCP. Это может оставаться проблемой, когда реализация MQTT 3.1.1 должна быть развернута в таких средах.

7.5 Получение сообщения

При получении Сервером права собственности на входящее сообщение приложения, он ДОЛЖЕН добавить его в состояние сеанса для тех Клиентов, которые имеют соответствующие Подписки. Правила соответствия определены в подразделе 7.7 [MQTT-7.5.0-1].

Обычно Клиенты получают сообщения в ответ на созданные ими Подписки. Клиент также может получать сообщения, которые не соответствуют ни одной из его прямых Подписок. Это может произойти, если Сервер автоматически назначил подписку Клиенту. Клиент также может получать сообщения, пока выполняется операция UNSUBSCRIBE. Клиент ДОЛЖЕН подтвердить любой пакет PUBLISH, который он получает, в соответствии с уровнями качества обслуживания доставки этого пакета, независимо от того, будет ли он обрабатывать содержащееся сообщение приложения. [MQTT-7.5.0-2].

7.6 Порядок отправки сообщений

Клиент ДОЛЖЕН следовать этим правилам при реализации потоков данных, определенных в данном разделе настоящего стандарта:

- когда он повторно отправляет какие-либо пакеты PUBLISH, повторно отправить их в том порядке, в котором были отправлены исходные пакеты PUBLISH (это относится к сообщениям QoS 1 и QoS 2) [MQTT-7.6.0-1];
- отправлять пакеты PUBACK в том порядке, в котором были получены соответствующие пакеты PUBLISH (сообщения QoS 1) [MQTT-7.6.0-2];

- отправлять пакеты PUBREC в том порядке, в котором были получены соответствующие пакеты PUBLISH (сообщения QoS 2) [MQTT-7.6.0-3];

- отправлять пакеты PUBREL в том порядке, в котором были получены соответствующие пакеты PUBREC (сообщения QoS 2) [MQTT-7.6.0-4].

Сервер ДОЛЖЕН по умолчанию считать каждую тему упорядоченной темой. Он МОЖЕТ предоставить административный или иной механизм, позволяющий считать одну или несколько тем упорядоченными темами [MQTT-7.6.0-5].

При обработке Сервером сообщения, опубликованного в упорядоченной теме, он ДОЛЖЕН следовать приведенным выше правилам при доставке сообщений каждому из своих подписчиков. Кроме того, он ДОЛЖЕН отправлять пакеты PUBLISH Клиентам (в этой же теме и с тем же уровнем качества обслуживания) в том порядке, в котором они были получены от любого указанного Клиента [MQTT-7.6.0-6].

Примечание — Правила, перечисленные выше, гарантируют, что, когда поток сообщений публикуется и на него оформляется подписка с уровнем качества обслуживания 1, окончательные копии каждого сообщения, передаваемого подписчикам, будут доставлены в том порядке, в котором они были первоначально опубликованы, но возможность дублирования сообщений может привести к в повторной отправке более раннего сообщения, полученного после одного из его последующих сообщений. Например, автор может отправлять сообщения в порядке 1, 2, 3, 4, а абонент может их получить в порядке 1, 2, 3, 2, 3, 4.

Если Клиент и Сервер удостоверяются, что в процессе доставки находится только одно сообщение в любой момент времени (не отправляя сообщение до тех пор, пока доставка предыдущего сообщения не будет подтверждена), то после любого последующего сообщения не будет получено пересылаемое ранее сообщение QoS 1 — например, абонент может получить их в порядке 1,2,3,3,4, но не 1,2,3,2,3,4. Ограничение размера потока пересылаемых сообщений до 1 сообщения также означает, что порядок будет сохранен, даже если автор отправляет последовательность сообщений с разными уровнями QoS по одной и той же теме.

7.7 Названия тем и фильтры тем

7.7.1 Подстановочные знаки в названиях тем

Разделитель уровня темы используется для введения структуры в название темы. Если он присутствует, он делит название темы на несколько «тематических уровней».

Фильтр тем подписки может содержать специальные символы подстановки, которые позволяют сразу подписываться на несколько тем.

Символы подстановки могут использоваться в фильтрах тем, но НЕ ДОЛЖНЫ использоваться в названии темы [MQTT-7.7.1-1].

7.7.1.1 Разделитель уровня

Прямая косая черта ('/' U + 002F) используется для разделения каждого уровня в дереве тем и обеспечения иерархической структуры названий тем. Использование разделителя уровня тем является важным, когда один из двух символов подстановки встречается в фильтрах тем, указанных Клиентами-подписчиками. Разделители тематических уровней могут отображаться в любом месте в фильтре тем или названии темы. Смежные разделители уровня тем указывают на уровень темы нулевой длины.

7.7.1.2 Многоуровневый шаблон

Знак решетки ('#' U + 0023) является подстановочным символом, который соответствует любому количеству уровней внутри темы. Многоуровневый подстановочный знак представляет собой родительский и любое количество дочерних уровней. Многоуровневый подстановочный знак ДОЛЖЕН быть указан либо самостоятельно, либо после разделителя уровня тем. В любом случае, он ДОЛЖЕН быть последним символом, указанным в фильтре тем [MQTT-7.7.1-2].

Пример — Если Клиент подписывается на «спорт/теннис/ теннисист1/#», он будет получать сообщения, опубликованные с использованием таких названий тем:

- «Спорт/теннис/теннисист1»;

- «Спорт/теннис/теннисист1/рейтинг»;

- «Спорт/теннис/теннисист1/счет/Уимблдон».

Кроме этого:

- «спорт/#» также соответствует единственному «спорту», так как # включает родительский уровень;

- «#» действителен и будет получать каждое сообщение приложения;

- «Спорт/теннис/#» действителен;

- *«Спорт/теннис#» недействителен;*
- *«Спорт/теннис/#/рейтинг» недействителен.*

7.7.1.3 Подстановочный шаблон одного уровня

Знак плюса (‘+’ U + 002B) является подстановочным символом, который соответствует только одному тематическому уровню.

Одноуровневый шаблон можно использовать на любом уровне фильтра тем, включая первый и последний уровни. Если он используется, он ДОЛЖЕН занимать весь уровень фильтра [MQTT-7.7.1-3]. Он может использоваться на нескольких уровнях в фильтре тем, и может использоваться в сочетании с многоуровневым шаблоном.

Пример — «спорт/теннис/+» соответствует «Спорт/теннис/ теннисист1» и «Спорт/теннис/ теннисист2», но не «Спорт/теннис/ теннисист1/рейтинг». Кроме того, поскольку одноуровневый шаблон соответствует только одному уровню, «Спорт/+» не соответствует «спорт», но он соответствует «спорт/».

Кроме этого:

- *«+» действителен;*
- *«+/Теннис/#» действителен;*
- *«Спорт+» недействителен;*
- *«Спорт/+/теннисист1» действителен;*
- *«/финансы» соответствует «+/+» и «/+», но не «+».*

7.7.2 Темы, начинающиеся с $

Сервер НЕ ДОЛЖЕН отождествлять фильтры тем, которые начинаются с символа подстановки (# или +) с названиями тем, начинающимися с символа $ [MQTT-7.7.2-1]. Серверу СЛЕДУЕТ запрещать Клиентам использовать такие имена тем для обмена сообщениями с другими Клиентами. Реализации Сервера МОГУТ использовать названия тем, начинающиеся с символа «$» для других целей.

Примечания

- $SYS/ широко используется в качестве префикса для тем, которые содержат информацию о конкретном Сервере или управляющую информацию API;
- Приложения не могут использовать тему с начальным символом $ для своих целей;
- Подписка на «#» не позволит получать сообщения, опубликованные в теме, начинающейся с «$»;
- Подписка на «+/мониторинг/Клиенты» не позволит получать сообщения, опубликованные в теме «$SYS/мониторинг/Клиенты»
- Подписка на «$SYS/#» позволит получать сообщения, опубликованные по темам, начинающимся с «$ SYS/»
- Подписка на «$SYS/мониторинг+» позволит получать сообщения, опубликованные в темах «$SYS/мониторить/Клиенты»

Для получения сообщений по темам, начинающимся с $ SYS/ и по темам, которые не начинаются с $, Клиенту СЛЕДУЕТ подписаться на «#» и «$SYS/#»

7.7.3 Семантика темы и ее использование

К названиям тем и фильтров тем применяются следующие правила:

- все названия тем и фильтры тем ДОЛЖНЫ иметь, как минимум, один символ [MQTT-7.7.3-1];
- названия тем и фильтры тем чувствительны к регистру;
- имена тем и фильтры темы могут включать символ пробела;
- первый или конечный символ «/» создает отдельное название темы или фильтр тем;
- название темы или фильтр тем, состоящий только из символа «/», действителен;
- название темы и фильтры тем НЕ ДОЛЖНЫ включать нулевой символ (Unicode U + 0000) [5] [MQTT-7.7.3-2];
- название темы и фильтры темы — это кодированные строки UTF-8, они НЕ ДОЛЖНЫ кодировать более 65535 байт [MQTT-7.7.3-3]. См. пункт 7.5.3.

Нет ограничений на количество уровней в названии темы или фильтре по темам, кроме тех, которые накладываются общей длиной кодированной строки UTF-8.

При сопоставлении подписок Сервер НЕ ДОЛЖЕН выполнять какую-либо нормализацию названий тем или фильтров тем, а также любую модификацию или замену нераспознанных символов [MQTT-7.7.3-4]. Каждый уровень в фильтре тем, за исключением содержащих символы подстановки, должен посимвольно совпадать с соответствующим уровне в названии темы для успешного установления совпадения.

Примечание — Правила кодирования UTF-8 означают, что сравнение фильтра темы и названия темы может быть выполнено либо путем сравнения кодированных байтов UTF-8, либо путем сравнения декодированных символов Юникода.

- «АККАУНТЫ» и «Аккаунты» — это два разных названия тем;
- «Кредиторская задолженность» — это действительное название темы;
- «/финансы» отличается от «финансы».

Сообщение приложения отправляется по каждой подписке Клиента, фильтр тем которой соответствует названию темы, прикрепленному к сообщению приложения. Ресурс темы МОЖЕТ быть либо предопределен на Сервере администратором, либо МОЖЕТ быть динамически создан Сервером, когда он получает первую подписку или сообщение приложения с этим именем темы. Сервер МОЖЕТ также использовать компонент безопасности для выборочной авторизации действий в ресурсе темы для конкретного Клиента.

7.8 Обработка ошибок

Если не указано иное, когда Сервер или Клиент сталкиваются с нарушением протокола, они ДОЛЖНЫ разорвать сетевое соединение, посредством которого был получен управляющий пакет, вызвавший нарушение протокола [MQTT-7.8.0-1].

В процессе работы Клиент или Сервер могут столкнуться с несистематической ошибкой (например, переполнение буфера), которая не дает возможности выполнить успешную обработку пакета MQTT.

Если Клиент или Сервер обнаруживают несистематическую ошибку при обработке входящего управляющего пакета, они ДОЛЖНЫ разорвать сетевое соединение, посредством которого получен данный управляющий пакет [MQTT-7.8.0-2]. Если Сервер обнаруживает несистематическую ошибку, ему НЕ СЛЕДУЕТ отсоединяться или оказывать какое-либо другое влияние на его взаимодействие с любым другим Клиентом.

8 Использование WebSocket в качестве протокола передачи данных

В случае, если управляющие пакеты MQTT передаются поверх протокола WebSocket [4], применяются следующие условия:

- управляющие пакеты MQTT ДОЛЖНЫ быть помещены в бинарные пакеты данных WebSocket. Если получен какой-либо другой тип пакета данных, получатель ДОЛЖЕН прервать сетевое соединение [MQTT-8.0.0-1];
- один пакет данных WebSocket может содержать как множество управляющих пакетов, так и лишь часть одного управляющего пакета MQTT. Получатель НЕ ДОЛЖЕН предполагать, что управляющие пакеты MQTT соответствуют рамкам пакета данных WebSocket [MQTT-8.0.0-2];
- Клиент ДОЛЖЕН включать ключевое слово «mqtt» в список суб-протоколов WebSocket, который он предлагает [MQTT-8.0.0-3];
- имя субпротокола WebSocket, выбранное и возвращаемое Сервером, должно быть «mqtt» [MQTT-8.0.0-4];
- URI WebSocket, используемый для подключения Клиента и Сервера, не влияет на протокол MQTT.

8.1 Примечания IANA

В настоящем стандарте IANA регистрирует суб-протокол WebSocket MQTT в реестре подпрограммного имени WebSocket Name со следующими данными, приведенными в таблице 56.

Таблица 56 — Идентификатор IANA WebSocket

Часть суб-протокола	Данные регистрации
Идентификатор суб-протокола	Mqtt
Общее имя суб-протокола	Mqtt
Определение суб-протокола	gost.e-ecolog.ru .html

9 Совместимость

Спецификация MQTT определяет совместимость реализаций Клиента MQTT и реализации Сервера MQTT.

Положения стандарта MQTT МОГУТ использоваться для реализации как Клиента, так и Сервера MQTT. Сервер, который принимает входящие подключения и устанавливает исходящие подключения к другим Серверам, ДОЛЖЕН быть совместим как с Клиентами MQTT, так и с Серверами MQTT [MQTT-9.0.0-1].

Совместимые реализации НЕ ДОЛЖНЫ требовать использования каких-либо расширений, определенных вне настоящего стандарта, чтобы взаимодействовать с любой другой совместимой разработкой [MQTT-9.0.0-2].

9.1 Цели совместимости

9.1.1 Сервер MQTT

Сервер MQTT соответствует настоящему стандарту, только если он удовлетворяет всем приведенным ниже инструкциям:

а) Формат всех управляющих пакетов, отправляемых Сервером, соответствует формату, описанному в разделах 5 и 6;

б) Он соответствует правилам сопоставления тем, описанным в подразделе 7.7;

в) Он соответствует всем требованиям уровня MUST (ДОЛЖЕН) в следующих разделах, которые указаны ниже, за исключением тех, которые применимы только к Клиенту:

- Раздел 4 — Представление данных;
- Раздел 5 — Формат управляющего пакета MQTT;
- Раздел 6 — Управляющие пакеты MQTT;
- Раздел 7 — Описание работы протоколов;
- Раздел 8 — Использование WebSocket в качестве протокола передачи данных;
- Раздел 9 — Совместимость.

Совместимый Сервер ДОЛЖЕН поддерживать использование одного или нескольких базовых транспортных протоколов, которые обеспечивают упорядоченный, не допускающий потерь поток байтов от Клиента к Серверу и от Сервера к Клиенту [MQTT-9.1.1-1]. Однако соответствие требованиям стандарта не зависит от выполнения им каких-либо конкретных транспортных протоколов. Сервер МОЖЕТ поддерживать любой из транспортных протоколов, перечисленных в подразделе 7.2, или любой другой транспортный протокол, соответствующий требованиям [MQTT-9.1.1-1].

9.1.2 Клиент MQTT

Клиент MQTT соответствует этой спецификации, только если он удовлетворяет всем приведенным ниже инструкциям:

а) Формат всех управляющих пакетов, отправляемых Клиентом, соответствует формату, описанному в разделах 5 и 6.

б) Он удовлетворяет всем требованиям уровня MUST (ДОЛЖЕН) в следующих разделах, которые указаны, кроме тех, которые применяются только к Серверу:

- Раздел 4 — Представление данных;
- Раздел 5 — Формат управляющего пакета MQTT;
- Раздел 6 — Управляющие пакеты MQTT;
- Раздел 7 — Описание работы протоколов;
- Раздел 8 — Использование WebSocket в качестве протокола передачи данных;
- Раздел 9 — Совместимость.

Совместимый Клиент ДОЛЖЕН поддерживать использование одного или нескольких базовых транспортных протоколов, которые обеспечивают упорядоченный, не допускающий потерь поток байтов от Клиента к Серверу и от Сервера к Клиенту [MQTT-9.1.2-1]. Однако соответствие требованиям настоящего стандарта не зависит от выполнения им каких-либо конкретных транспортных протоколов. Клиент МОЖЕТ поддерживать любой из транспортных протоколов, перечисленных в подразделе 7.2, или любой другой транспортный протокол, соответствующий требованиям [MQTT-9.1.2-1].

Приложение А
(рекомендуемое)

Безопасность

А.1 Введение

Данное приложение предоставляется только в качестве справочного материала и является **ненормативным**. Тем не менее, по настоятельной рекомендации, приведенной в настоящем стандарте, в реализациях Сервера, использующих протокол TLS, использовался TCP-порт 8883 (название службы IANA: secure-mqtt).

Существует ряд угроз, которые следует учитывать разработчикам, например:

- устройства могут быть дефектны;
- данные, хранящиеся у Клиентов и на Серверах, могут быть доступны;
- поведение протокола может иметь побочные эффекты (например, «временные атаки»);
- атаки, вызывающие отказ в обслуживании (DoS);
- связь может быть перехвачена, изменена, перенаправлена или раскрыта;
- в поток данных могут быть внедрены поддельные управляющие пакеты.

Решения MQTT часто развертываются во враждебных средах связи. В таких случаях реализациям часто необходимо предусмотреть механизмы для:

- идентификации пользователей и устройств;
- авторизации доступа к ресурсам Сервера;
- целостности управляющих пакетов MQTT и данных приложения, содержащихся в них;
- конфиденциальности управляющих пакетов MQTT и данных приложения, содержащихся в них.

В качестве транспортного протокола MQTT занимается только передачей сообщений, и ответственность за обеспечение безопасности лежит на разработчике. Для этого обычно применяется протокол TLS [3].

В дополнение к техническим проблемам безопасности также могут существовать географические (например, «зона безопасности» США-Европа [25]), отраслевые (например, PCI DSS [16]) и регулятивные (например, Sarbanes-Oxley [24]).

А.2 Решения MQTT: безопасность и сертификация

Реализация может потребовать соответствия определенным отраслевым стандартам безопасности, таким как NIST Cyber Security Framework (Инфраструктура кибербезопасности [13], PCI-DSS [16]), FIPS-140-2 [9] и NSA Suite B [15].

Руководство по использованию MQTT в рамках NIST Cyber Security Framework [13] можно найти в дополнительных публикациях MQTT, MQTT и NIST Framework for Improving Critical Infrastructure Cybersecurity (Структура для улучшения критической инфраструктуры кибербезопасности) [12]. Использование проверенных в отрасли, независимо проверенных и сертифицированных технологий поможет соответствовать требованиям.

А.3 Упрощенная криптография и ограниченные устройства

Широко применяется Улучшенный стандарт шифрования (AES) [7] и Стандарт шифрования данных (DES) [8].

ISO 29192 [11] дает рекомендации в отношении криптографических примитивов, специально настроенных для работы на устройствах «низкого уровня» с ограниченными вычислительными мощностями.

А.4 Замечания по реализации

При внедрении или использовании MQTT необходимо учитывать многие проблемы безопасности. Следующий раздел не следует рассматривать как прямое руководство.

Реализация может выполнять все или некоторые из приведенных ниже требований:

А.4.1 Идентификация Клиентов Сервером

Пакет CONNECT содержит поля «Имя пользователя» и «Пароль». При реализации можно выбирать, как использовать содержимое этих полей. Они могут предоставить свой собственный механизм идентификации, использовать внешнюю систему идентификации, такую как токены LDAP [18] или OAuth [22], или использовать механизмы идентификации операционной системы.

Реализации, передающие данные идентификации с помощью открытого текста, игнорирующие эти параметры или требующие отсутствия данных идентификации, должны знать, что это может привести к атакам, связанным с подключением взломщика к каналу связи и взломом защиты путем замещения оригинала. В подразделе 5.4.5 описаны способы обеспечения конфиденциальности данных.

Виртуальная частная сеть (VPN) между Клиентами и Серверами может обеспечить уверенность в том, что данные принимаются только от авторизованных Клиентов.

Если используется TLS [3], SSL-сертификаты, отправленные от Клиента, могут использоваться Сервером для идентификации Клиента.

Реализация может допустить идентификацию, когда учетные данные отправляются в сообщении приложения Клиентом на Сервер.

А.4.2 Авторизация Клиентов Сервером

Реализация может ограничивать доступ к ресурсам Сервера на основе информации, предоставленной Клиентом, такой как Имя пользователя, Идентификатор Клиента, имя хоста/IP-адрес Клиента или результат механизмов аутентификации.

А.4.3 Аутентификация Сервера Клиентом

Протокол MQTT не является надежным симметричным протоколом: он не предоставляет механизм для проверки подлинности Сервера.

Если используется TLS [3], SSL-сертификаты, отправленные с Сервера, могут использоваться Клиентом для идентификации Сервера. Реализации, предоставляющие услугу MQTT для нескольких имен хостов с одного IP-адреса, должны быть осведомлены о расширении указателя имени Сервера для TLS, определенных в разделе 3 RFC [21]. Это позволяет Клиенту сообщать Серверу имя хоста Сервера, к которому он пытается подключиться.

Реализация может позволить идентификацию, когда учетные данные отправляются в сообщении приложения от Сервера Клиенту.

VPN между Клиентами и Серверами может обеспечить уверенность в том, что Клиенты подключаются к предполагаемому Серверу.

А.4.4 Целостность сообщений приложений и управляющих пакетов

Приложения могут независимо включать хэш-значения в свои сообщения приложений. Это может обеспечить целостность содержимого управляющих пакетов PUBLISH.

TLS [3] предоставляет хэш-алгоритмы для проверки целостности данных, передаваемых по сети.

Использование VPN для подключения Клиентов и Серверов может обеспечить целостность данных в разделе сети, охватываемой VPN.

А.4.5 Конфиденциальность сообщений приложений и управляющих пакетов

TLS [3] может обеспечивать шифрование данных, передаваемых по сети. Имеются действительные пакеты шифрования TLS, которые включают алгоритм шифрования NULL, который не шифрует данные. Чтобы обеспечить конфиденциальность Клиенты и Серверы должны избегать этих наборов шифров.

Приложение может самостоятельно шифровать содержимое своих сообщений приложения. Это может обеспечить конфиденциальность сообщения приложения как по сети, так и в состоянии покоя. Это не обеспечило бы конфиденциальность для других свойств сообщения приложения, например, названия темы.

Разработки Клиента и Сервера могут предоставлять зашифрованное хранилище для данных в состоянии покоя, таких как сообщения приложений, хранящиеся как часть Сеанса.

Использование VPN для подключения Клиентов и Серверов может обеспечить конфиденциальность данных в разделе сети, охватываемом VPN.

А.4.6 Отказ от передачи сообщений

Разработчики приложений, возможно, должны будут рассмотреть соответствующие стратегии для достижения полной невозможности отказа от факта передачи по сети.

А.4.7 Нахождение компромисса между Клиентами и Серверами

Реализации Клиента и Сервера с использованием TLS [3] должны обеспечивать возможность того, чтобы любые SSL-сертификаты, предоставляемые при запуске соединения TLS [3], были связаны с именем хоста подключающегося Клиента, или с Сервером, к которому производится подключение.

Реализации Клиента и Сервера с использованием TLS [3] могут предоставлять возможности проверки списков отзыва сертификатов (CRLs [20]) и онлайн протокола статуса сертификатов (OSCP) [23] для предотвращения использования отозванных сертификатов.

Физическое развертывание может сочетать защищенное от несанкционированного доступа оборудование с передачей конкретных данных в приложениях. Например, счетчик может иметь встроенный GPS, чтобы гарантировать, что он не используется в несанкционированном расположении. [10] является стандартом для реализации механизмов аутентификации идентификатора устройства с использованием криптографически привязанного идентификатора.

А.4.8 Обнаружение аномального поведения

Реализации Сервера могут отслеживать поведение Клиента для обнаружения потенциальных инцидентов безопасности. Например:

- повторные попытки подключения;
- повторные попытки аутентификации;
- некорректное прекращение соединений;
- сканирование темы (попытки отправить или подписаться на многие темы);
- отправка сообщений, доставка которых невозможна (без подписчиков на темы);
- подключение Клиентов, не отправляющих данные.

Реализации Сервера могут отключить Клиентов, которые нарушают его правила безопасности.

Реализации Сервера, обнаруживающие нежелательное поведение, могут создавать динамические списки блокировок на основе таких идентификаторов, как IP-адрес или идентификатор Клиента.

Реализации могут использовать средства управления сетевым уровнем (где доступно) для ограничения или блокировки скорости на основе IP-адреса или другой информации.

А.4.9 Другие соображения безопасности

Если утеряны сертификаты SSL Клиентов или Серверов, или считается, что они могут быть испорчены, их следует аннулировать (используя CRL [20] и/или OSCP [23]).

Учетные данные для проверки подлинности Клиента или Сервера, такие как имя пользователя и пароль, которые потеряны или считаются испорченными, должны быть аннулированы и/или переизданы.

В случае продолжительных по времени соединений:

- реализации Клиента и Сервера с использованием TLS [3] должны давать возможность повторного обсуждения Сеанса, чтобы устанавливать новые криптографические параметры (заменять ключи сеанса, менять набор шифров, изменять учетные данные для проверки подлинности);
- Серверы могут отключать Клиентов и требовать от них повторной идентификации с новыми учетными данными.

Устройства с ограниченными вычислительными мощностями и Клиенты в ограниченных сетях могут использовать возобновление сеанса TLS [19], чтобы снизить затраты на повторное подключение сеансов TLS [3].

Клиенты, подключенные к Серверу, имеют транзитивные доверительные отношения с другими Клиентами, подключенными к одному и тому же Серверу, и у которых есть полномочия публиковать данные по тем же темам.

А.4.10 Использование Серверов SOCKS

Реализация Клиентов должна учитывать, что в некоторых средах требуется использование прокси-серверов SOCKSv5 [17] для создания исходящих сетевых подключений. Некоторые реализации MQTT могут использовать альтернативные защищенные туннели (например, SSH) с использованием прокси-серверов SOCKS. В тех случаях, когда разработки предпочитают использовать SOCKS, они должны выполнять оба типа (анонимный и с паролем, подтверждающим имя пользователя), идентификации прокси-сервера SOCKS. В последнем случае реализации должны быть осведомлены о том, что идентификация SOCKS может возникать в текстовом виде, поэтому следует избегать использования тех же учетных данных для подключения к Серверу MQTT.

А.4.11 Профили безопасности

Исполнители и разработчики решений могут отнестись к безопасности как к набору профилей, которые могут применяться к протоколу MQTT. Ниже приведен пример иерархии уровней многоуровневой безопасности.

А.4.11.1 Чистый профиль коммуникации

При использовании чистого профиля коммуникации протокол MQTT работает по открытой сети без каких-либо дополнительных защищенных механизмов связи.

А.4.11.2 Защищенный профиль сетевой связи

При использовании профиля защищенной сетевой связи протокол MQTT проходит через физическую или виртуальную сеть, которая имеет элементы управления безопасностью, например, VPN или физически защищенную сеть.

А.4.11.3 Защищенный профиль транспортного протокола

При использовании защищенного профиля транспортный протокол MQTT проходит через физическую или виртуальную сеть и использует TLS [3], что обеспечивает идентификацию, целостность и конфиденциальность.

TLS [3] Идентификация Клиента может использоваться в дополнение к идентификации Клиента MQTT или вместо нее, как указано в полях Имя пользователя и Пароль.

А.4.11.4 Отраслевые профили безопасности

Предполагается, что протокол MQTT будет встроен в отраслевые профили приложений, каждый из которых определяет модель угрозы и конкретные механизмы безопасности, которые будут использоваться для устранения этих угроз. Рекомендации по конкретным механизмам безопасности часто принимаются на основании существующих документов, в том числе:

- Инфраструктура кибербезопасности NIST [13];
- NISTIR 7628 Рекомендации по обеспечению безопасности интеллектуальных сетей [14];
- Требования безопасности для криптографических модулей (FIPS PUB 140-2) [9];
- Стандарт безопасности данных для платежных карт PCI-DSS PCI [16];
- NSA Криптография, кат. B [15].

Приложение Б
(справочное)

Обязательные нормативные положения

Данное приложение предоставляется только в качестве справочного материала и является **ненормативным.** В таблице Б.1 приведен перечень заявлений о совместимости, содержащихся в основной части настоящего стандарта. Окончательный список требований к соответствию приведен в разделе 9.

Таблица Б.1 — Перечень нормативных положений о совместимости

Номер нормативного положения	Нормативное положение
[MQTT-4.5.3-1]	Символьные данные в кодированной строке UTF-8 ДОЛЖНЫ быть правильно сформированными UTF-8, как определено спецификацией Unicode [5] и пересчитаны в RFC 3629 [2]. В частности, эти данные НЕ ДОЛЖНЫ включать кодировки кодовых точек между U+D800 и U+DFFF. Если Сервер или Клиент получает управляющий пакет, содержащий неправильно сформированный UTF-8, он ДОЛЖЕН прервать сетевое подключение
[MQTT-4.5.3-2]	Закодированная строка UTF-8 НЕ ДОЛЖНА включать кодировку нулевого символа U+0000. Если получатель (Сервер или Клиент) получает управляющий пакет, содержащий U+0000, он ДОЛЖЕН прервать сетевое соединение
[MQTT-4.5.3-3]	Кодированная последовательность UTF-8 0xEF 0xBB 0xBF всегда должна интерпретироваться как означающая U+FEFF («Нулевая ширина без пробела»), где бы она ни появлялась в строке, и НЕ ДОЛЖНА быть пропущена или отключена получателем пакетов
[MQTT-5.2.2-1]	Если бит флага отмечен как «Зарезервировано» в таблице 5.2 — Биты флагов, он зарезервирован для будущего использования и ДОЛЖЕН быть установлен в значение, указанное в этой таблице
[MQTT-5.2.2-2]	Если получены недопустимые флаги, получатель ДОЛЖЕН прервать сетевое соединение
[MQTT-5.3.1-1]	SUBSCRIBE, UNSUBSCRIBE и PUBLISH (в случаях, когда QoS > 0) управляющие пакеты ДОЛЖНЫ содержать ненулевой 16-разрядный идентификатор пакета
[MQTT-5.3.1-2]	Каждый раз, когда Клиент отправляет новый пакет одного из этих типов, он ДОЛЖЕН назначить ему неиспользуемый в настоящее время пакетный идентификатор
[MQTT-5.3.1-3]	Если Клиент повторно отправляет конкретный управляющий пакет, он ДОЛЖЕН использовать тот же идентификатор пакета в последующих повторных отправках этого пакета. Идентификатор пакета становится доступным для повторного использования после того, как Клиент обработал соответствующий пакет подтверждения. В случае QoS 1 PUBLISH это соответствующий PUBACK; в случае QoS 2 это PUBCOMP. Для SUBSCRIBE или UNSUBSCRIBE — это соответствующий SUBACK или UNSUBACK
[MQTT-5.3.1-4]	Те же условия применяются к Серверу, когда он отправляет ОПУБЛИКОВАТЬ /PUBLISH/ с QoS > 0
[MQTT-5.3.1-5]	Пакет PUBLISH НЕ ДОЛЖЕН содержать идентификатор пакета, если его значение QoS установлено на 0
[MQTT-5.3.1-6]	Пакет PUBACK, PUBREC или PUBREL ДОЛЖЕН содержать тот же идентификатор пакета, что и первоначально отправленный пакет PUBLISH
[MQTT-5.3.1-7]	Аналогично SUBACK и UNSUBACK ДОЛЖНЫ содержать идентификатор пакета, который использовался в соответствующем пакете SUBSCRIBE и UNSUBSCIBE, соответственно
[MQTT-6.1.0-1]	После того как Клиент установил сетевое соединение с Сервером, первым пакетом, отправленным Клиентом на Сервер, ДОЛЖЕН быть пакет CONNECT
[MQTT-6.1.0-2]	Сервер ДОЛЖЕН обработать второй пакет CONNECT, отправленный Клиентом, как нарушение протокола, и отключить Клиента

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-6.1.2-1]	Если Имя протокола неверно, Сервер МОЖЕТ отключить Клиента, или МОЖЕТ продолжить обработку пакета CONNECT в соответствии с некоторыми другими спецификациями. В последнем случае Сервер НЕ ДОЛЖЕН продолжать обрабатывать пакет CONNECT в соответствии с настоящей спецификацией
[MQTT-6.1.2-2]	Сервер ДОЛЖЕН реагировать на пакет CONNECT передачей кода CONNACK 0x01 (неприемлемый уровень протокола), а затем отключает Клиента, если уровень протокола не поддерживается Сервером
[MQTT-6.1.2-3]	Сервер ДОЛЖЕН подтвердить, что зарезервированный флаг в пакете управления CONNECT установлен на ноль и прервать соединение с Клиентом, если значение не равно нулю
[MQTT-6.1.2-4]	Если для параметра «Очистить сеанс» установлено значение «0», Сервер ДОЛЖЕН возобновить обмен данными с Клиентом на основе состояния с текущего сеанса (как определено идентификатором Клиента). Если сеанс не связан с идентификатором Клиента, Сервер ДОЛЖЕН создать новый сеанс. Клиент и Сервер ДОЛЖНЫ сохранять сеанс после разъединения Клиента и Сервера
[MQTT-6.1.2-5]	После отключения Сеанса, по которому параметром «Очистить сеанс» установлен на 0, Сервер ДОЛЖЕН сохранить дополнительные сообщения QoS 1 и QoS 2, соответствующие любым подпискам, которые существовали у Клиента во время отключения в составе состояния Сеанса
[MQTT-6.1.2-6]	Если для параметра «Очистить сеанс» установлено значение 1, Клиент и Сервер ДОЛЖНЫ отменить предыдущий Сеанс и запустить новый. Этот Сеанс длится до тех пор, пока существует сетевое подключение. Данные состояния, связанные с этим Сеансом, НЕ ДОЛЖНЫ повторно использоваться в любой последующей сессии
[MQTT-6.1.2.7]	Сохраненные сообщения не являются частью состояния Сеанса на Сервере, они НЕ ДОЛЖНЫ удаляться, когда Сеанс заканчивается
[MQTT-6.1.2-8]	Если флаг дальнейших действий установлен на 1, это означает, что, если запрос о соединении принят, Сообщение о дальнейших действиях ДОЛЖНО храниться на Сервере и связано с сетевым подключением. Сообщение о дальнейших действиях ДОЛЖНО быть опубликовано, когда Сетевое подключение будет закрыто, если сообщение не было удалено Сервером при получении пакета DISCONNECT
[MQTT-6.1.2-9]	Если флаг дальнейших действий установлен на 1, поля будущего QoS и будущего сохранения в флагах CONNECT будут использоваться Сервером, а поля будущей темы и сообщение о дальнейших действиях ДОЛЖНЫ присутствовать в полезной нагрузке
[MQTT-6.1.2-10]	Сообщение о дальнейших действиях ДОЛЖНО быть удалено из сохраненного состояния Сеанса на Сервере после его публикации, или, когда Сервер получил от Клиента пакет DISCONNECT
[MQTT-6.1.2-11]	Если флаг дальнейших действий установлен на 0, значения полей QoS и будущего сохранения должны быть установлены равными нулю, а поля будущая тема и сообщение о дальнейших действиях НЕ ДОЛЖНЫ присутствовать в полезной нагрузке
[MQTT-6.1.2-12]	Если флаг дальнейших действий установлен на 0, сообщение о дальнейших действиях НЕ ДОЛЖНО публиковаться при завершении этого сетевого подключения
[MQTT-6.1.2-13]	Если флаг дальнейших действий установлен на 0, то будущий QoS ДОЛЖНО быть установлено на 0 (0x00)
[MQTT-6.1.2-14]	Если флаг дальнейших действий установлен на 1, значение будущего QoS может быть 0 (0x00), 1 (0x01) или 2 (0x02). Значение НЕ ДОЛЖНО быть установлено на 3 (0x03)
[MQTT-6.1.2-15]	Если флаг дальнейших действий установлен на 0, то флаг будущего сохранения ДОЛЖЕН быть установлен на 0

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-6.1.2-16]	Если для будущего сохранения установлено значение 0, Сервер ДОЛЖЕН публиковать сообщение о дальнейших действиях как не сохраненное сообщение
[MQTT-6.1.2-17]	Если для будущего сохранения установлено значение 1, Сервер ДОЛЖЕН публиковать сообщение о дальнейших действиях как сохраненное сообщение
[MQTT-6.1.2-18]	Если параметр флага с именем пользователя установлен на 0, имя пользователя НЕ ДОЛЖНО присутствовать в полезной нагрузке
[MQTT-6.1.2-19]	Если параметр флага с именем пользователя установлен на 1, имя пользователя ДОЛЖНО присутствовать в полезной нагрузке
[MQTT-6.1.2-20]	Если флаг пароля установлен на 0, пароль НЕ ДОЛЖЕН присутствовать в полезной нагрузке
[MQTT-6.1.2-21]	Если флаг пароля установлен на 1, пароль ДОЛЖЕН присутствовать в полезной нагрузке
[MQTT-6.1.2-22]	Если параметр флага с именем пользователя установлен на 0, флаг пароля ДОЛЖЕН быть установлен на 0
[MQTT-6.1.2-23]	Клиент несет ответственность за то, чтобы промежуток между отправляемыми управляющими пакетами не превышал значение активного соединения. В случае отсутствия отправки каких-либо других управляющих пакетов Клиент ДОЛЖЕН отправить пакет PINGREQ
[MQTT-6.1.2-24]	Если значение активного соединения не равно нулю, и Сервер не получает контрольный пакет от Клиента в течение полутора интервалов периода активного соединения, он ДОЛЖЕН прервать сетевое подключение с Клиентом, как если бы произошел сбой сети
[MQTT-6.1.3-1]	Полезная нагрузка пакета CONNECT содержит одно или несколько полей с префиксом длины, наличие которых определяется флагами в заголовке переменной. Эти поля, если они есть, ДОЛЖНЫ появляться в порядке: идентификатора Клиента, будущая тема, сообщение о дальнейших действиях, Имя пользователя, пароль
[MQTT-6.1.3-2]	Каждый Клиент, подключающийся к Серверу, имеет уникальный ClientId. Идентификатор Клиента ДОЛЖЕН использоваться Клиентами и Серверами для определения их состояния в отношении этого сеанса MQTT между Клиентом и Сервером
[MQTT-6.1.3-3]	Идентификатор Клиента (ClientId) ДОЛЖЕН присутствовать и ДОЛЖЕН быть первым полем в полезной нагрузке пакета CONNECT
[MQTT-6.1.3-4]	ClientId ДОЛЖЕН быть кодированной строкой UTF-8, как определено в пункте 4.5.3 [MQTT-6.1.3-4]. Сервер ДОЛЖЕН допускать ClientIds, длина которых попадает в диапазон 1—23 кодированных байтов UTF-8, и содержит только символы «0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ»
[MQTT-6.1.3-6]	Сервер МОЖЕТ позволить Клиенту предоставить ClientId, длина которого равна нулю, однако, если он делает это, Сервер ДОЛЖЕН рассматривать это как особый случай и назначить уникальный ClientId для этого Клиента. Он ДОЛЖЕН обрабатывать пакет CONNECT, как если бы Клиент предоставил уникальный ClientId
[MQTT-6.1.3-7]	Если Клиент отправляет ClientId с нулевым байтом, Клиент ДОЛЖЕН также установить «Очистить сеанс» на 1
[MQTT-6.1.3-8]	Если Клиент отправляет ClientId с нулевым байтом, с установленным значением «Очистить сеанс» равным 0, Сервер ДОЛЖЕН отвечать на пакет CONNECT кодом возврата CONNACK 0x02 (отклонение идентификатора), а затем прервать сетевое соединение
[MQTT-6.1.3-9]	Если Сервер отклоняет ClientId, он ДОЛЖЕН отвечать на пакет CONNECT кодом возврата CONNACK 0x02 (идентификатор отклонен), а затем прервать сетевое соединение

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-6.1.3-10]	Если флаг дальнейших действий установлен на 1, будущая тема будет следующим полем в полезной нагрузке. Будущая тема ДОЛЖНА быть кодированной строкой UTF-8, как определено в пункте 4.5.3
[MQTT-6.1.3-11]	Имя пользователя ДОЛЖНО быть кодированной строкой UTF-8, как определено в пункте 4.5.3
[MQTT-6.1.4-1]	Сервер ДОЛЖЕН подтвердить, что пакет CONNECT соответствует Разделу 6.1, и прервать сетевое соединение, не отправляя CONNACK, если он не соответствует
[MQTT-6.1.4-2]	Если ClientId представляет Клиента, уже подключенного к Серверу, Сервер ДОЛЖЕН отключить существующего Клиента
[MQTT-6.1.4-3]	Сервер ДОЛЖЕН выполнить обработку «Очистить сеанс», описанного в подпункте 6.1.2.4
[MQTT-6.1.4-4]	Сервер ДОЛЖЕН подтвердить пакет CONNECT с помощью пакета CONNACK, содержащего нулевой код возврата
[MQTT-6.1.4-5]	Если Сервер отклоняет CONNECT, он НЕ ДОЛЖЕН обрабатывать любые данные, отправленные Клиентом после пакета CONNECT
[MQTT-6.2.0-1]	Первый пакет, отправленный с Сервера Клиенту, ДОЛЖЕН быть пакетом CONNACK
[MQTT-6.2.2-1]	Если Сервер принимает соединение с установкой параметра «Очистить сеанс» на 1, Сервер ДОЛЖЕН установить текущий сеанс на 0 в пакете CONNACK в дополнение к установке нулевого кода возврата в пакете CONNACK
[MQTT-6.2.2-2]	Если Сервер принимает соединение с установкой параметра «Очистить сеанс» на 0, значение, установленное для текущего сеанса, зависит от того, сохранил ли Сервер состояние сеанса для предоставленного идентификатора Клиента. Если Сервер сохранил состояние сеанса, он ДОЛЖЕН установить текущий сеанс на 1 в пакете CONNACK
[MQTT-6.2.2-3]	Если Сервер не сохранил состояние Сеанса, он ДОЛЖЕН установить текущий сеанс на 0 в пакете CONNACK. Это дополнительно к установке нулевого кода возврата в пакете CONNACK
[MQTT-6.2.2-4]	Если Сервер отправляет пакет CONNACK, содержащий ненулевой код возврата, он ДОЛЖЕН установить текущий сеанс на 0
[MQTT-6.2.2-5]	Если Сервер отправляет пакет CONNACK, содержащий ненулевой код возврата, он ДОЛЖЕН затем прервать сетевое соединение
[MQTT-6.2.2-6]	Если ни один из кодов возврата, перечисленных в таблице 6.1 — Значения кода возврата соединения, не считается применимым, тогда Сервер ДОЛЖЕН закрыть сетевое соединение, не отправляя пакет CONNACK
[MQTT-6.3.1-1]	Флаг DUP ДОЛЖЕН быть установлен на 1 Клиентом или Сервером, когда он пытается повторно отправить пакет PUBLISH
[MQTT-6.3.1-2]	Флаг DUP ДОЛЖЕН быть установлен на 0 для всех сообщений QoS 0
[MQTT-6.3.1-3]	Значение флага DUP из входящего пакета PUBLISH не распространяется, когда пакет PUBLISH отправляется подписчикам Сервером. Флаг DUP в исходящем пакете PUBLISH устанавливается независимо от входящего пакета PUBLISH, его значение ДОЛЖНО быть определено только исходя из того, является ли исходящий пакет PUBLISH повторной передачей
[MQTT-6.3.1-4]	Пакет PUBLISH НЕ ДОЛЖЕН иметь оба бита QoS, установленные на 1. Если Сервер или Клиент получает пакет PUBLISH, у которого оба бита QoS установлены на 1, он ДОЛЖЕН прервать сетевое подключение

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-6.3.1-5]	Если флаг RETAIN установлен на 1, в пакете PUBLISH, отправленном Клиентом на Сервер, Сервер ДОЛЖЕН хранить сообщение приложения и его QoS, чтобы он мог доставляться будущим подписчикам, подписки которых соответствуют их названию темы
[MQTT-6.3.1-6]	Когда установлена новая подписка, последнее сообщение с сохранением, если оно есть, по каждому имени соответствующего типа ДОЛЖНО быть отправлено подписчику
[MQTT-6.3.1-7]	Если Сервер получает сообщение QoS 0 с флагом RETAIN, установленным на 1, он ДОЛЖЕН отменить любое сообщение, ранее сохраненное для этой темы. СЛЕДУЕТ сохранить новое сообщение QoS 0 в качестве нового сохраненного сообщения для этой темы, но МОЖНО отказаться от него в любое время — если это произойдет, для этого раздела не будет сохраненного сообщения
[MQTT-6.3.1-8]	При отправке пакета PUBLISH Клиенту Сервер ДОЛЖЕН установить флаг RETAIN на 1, если сообщение отправлено в результате новой подписки, сделанной Клиентом
[MQTT-6.3.1-9]	Он ДОЛЖЕН установить флаг RETAIN на 0, когда пакет PUBLISH отправляется Клиенту, поскольку он соответствует установленной подписке, независимо от того, как был установлен флаг в полученном сообщении
[MQTT-6.3.1-10]	Пакет PUBLISH с флагом RETAIN, установленным на 1, и полезная нагрузка, содержащая нулевые байты, будет обрабатываться как обычно Сервером и отправляться Клиентам с подпиской, соответствующей названию темы. Кроме того, любое существующее сохраненное сообщение с тем же именем темы ДОЛЖНО быть удалено, и любые будущие подписчики для темы не получат сохраненное сообщение
[MQTT-6.3.1-11]	Сохраненное сообщение с нулевым байтом НЕ ДОЛЖНО храниться как сохраненное сообщение на Сервере
[MQTT-6.3.1-12]	Если флаг RETAIN равен 0, в пакете PUBLISH, отправленном Клиентом на Сервер, Сервер НЕ ДОЛЖЕН хранить это сообщение и НЕ ДОЛЖЕН удалять или заменять существующее сохраненное сообщение
[MQTT-6.3.2-1]	Название темы ДОЛЖНО присутствовать в качестве первого поля в заголовке пакета PUBLISH. Он ДОЛЖЕН быть кодированной кодировкой UTF-8
[MQTT-6.3.2-2]	Название темы в пакете PUBLISH НЕ ДОЛЖНО содержать подстановочных знаков
[MQTT-6.3.2-3]	Название темы в пакете PUBLISH, отправленное Сервером Клиенту-подписчику, должно соответствовать фильтру темы подписки в соответствии с процессом сопоставления, определенным в Разделе 4.7
[MQTT-6.3.4-1]	Получатель пакета PUBLISH ДОЛЖЕН отвечать в соответствии с таблицей 6.4 — Ожидаемый ответ пакета PUBLISH, как определено QoS в пакете PUBLISH
[MQTT-6.3.5-1]	Сервер ДОЛЖЕН доставлять сообщение Клиенту в соответствии с максимальным QoS всех соответствующих подписок
[MQTT-6.3.5-2]	Если реализация Сервера не разрешает Клиенту выполнять пакет PUBLISH; он не может сообщить об этом Клиенту. Он ДОЛЖЕН сделать положительное подтверждение в соответствии с обычными правилами QoS или прервать сетевое соединение
[MQTT-6.6.1-1]	Биты 3, 2, 1 и 0 фиксированного заголовка в пакете управления PUBREL зарезервированы и ДОЛЖНЫ быть установлены на 0, 0, 1 и 0 соответственно. Сервер ДОЛЖЕН считать любое другое значение искаженным, и прервать сетевое соединение
[MQTT-6.8.1-1]	Биты 3, 2, 1 и 0 фиксированного заголовка управляющего пакета SUBSCRIBE зарезервированы и ДОЛЖНЫ быть установлены на 0, 0, 1 и 0 соответственно. Сервер ДОЛЖЕН считать любое другое значение искаженным и прервать сетевое соединение
[MQTT-6.8.3-1]	Фильтры тем в полезной нагрузке пакета SUBSCRIBE ДОЛЖНЫ быть закодированными UTF-8 строками, как определено в Разделе 4.5.3

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-6.8.3-2]	Если он предпочитает не поддерживать фильтры тем, которые содержат подстановочные знаки, он ДОЛЖЕН отклонить любой запрос на подписку, фильтр которого их содержит
[MQTT-6.8.3-3]	Полезная нагрузка пакета SUBSCRIBE ДОЛЖНА содержать по меньшей мере один фильтр тем/пару QoS. Пакет SUBSCRIBE без полезной нагрузки является нарушением протокола
[MQTT-6-8.3-4]	Сервер ДОЛЖЕН считать пакет SUBSCRIBE искаженным и прервать сетевое соединение, если любой из зарезервированных битов в полезной нагрузке отличен от нуля, или QoS не равно 0, 1 или 2
[MQTT-6.8.4-1]	Когда Сервер получает пакет SUBSCRIBE от Клиента, Сервер ДОЛЖЕН отвечать с пакетом SUBACK
[MQTT-6.8.4-2]	Пакет SUBACK ДОЛЖЕН иметь тот же идентификатор пакета, что и пакет SUBSCRIBE, который он подтверждает
[MQTT-6.8.4-3]	Если Сервер получает пакет SUBSCRIBE, содержащий фильтр тем, который идентичен существующему фильтру тем Подписки, он ДОЛЖЕН полностью заменить существующую Подписку новой Подпиской. Фильтр темы в новой Подписке будет идентичен фильтру темы в предыдущей Подписке, хотя его максимальное значение QoS может отличаться. Любые существующие сохраненные сообщения, соответствующие фильтру темы, ДОЛЖНЫ быть повторно отправлены, но поток публикаций НЕ ДОЛЖЕН прерываться
[MQTT-6.8.4-4]	Если Сервер получает пакет SUBSCRIBE, который содержит несколько фильтров тем, он ДОЛЖЕН обрабатывать этот пакет, как если бы он получил последовательность из нескольких пакетов SUBSCRIBE, за исключением того, что он объединяет их ответы в один ответ SUBACK
[MQTT-6.8.4-5]	Пакет SUBACK, отправленный Сервером Клиенту, ДОЛЖЕН содержать код возврата для каждого фильтра тем/пары QoS. Этот код возврата ДОЛЖЕН показать максимальное QoS, которое было предоставлено для этой Подписки, или указать на сбой подписки
[MQTT-6.8.4-6]	Сервер может предоставить более низкий максимальный QoS, чем запросил подписчик. QoS сообщений полезной нагрузки, отправленных в ответ на подписку, ДОЛЖНЫ быть минимальным QoS изначально опубликованного сообщения и максимального QoS, предоставленного Сервером. Серверу разрешено отправлять дубликаты копий сообщения подписчику в случае, когда исходное сообщение было опубликовано с QoS 1, а максимальным предоставленным QoS был QoS 0
[MQTT-6.9.3-1]	Порядок кодов возврата в пакете SUBACK ДОЛЖЕН соответствовать порядку фильтров тем в пакете SUBSCRIBE
[MQTT-6.9.3-2]	Коды возврата SUBACK, отличные от 0x00, 0x01, 0x02 и 0x80, зарезервированы и НЕ ДОЛЖНЫ использоваться
[MQTT-6.10.1-1]	Биты 3, 2, 1 и 0 фиксированного заголовка управляющего пакета UNSUBSCRIBE зарезервированы и ДОЛЖНЫ быть установлены на 0, 0, 1 и 0 соответственно. Сервер ДОЛЖЕН обрабатывать любое другое значение как искаженное и закрывать сетевое соединение
[MQTT-6.10.3-1]	Фильтры тем в пакете UNSUBSCRIBE ДОЛЖНЫ быть закодированными строками UTF-8, как определено в разделе 4.5.3, формируя непрерывный пакет
[MQTT-6.10.3-2]	Фильтры тем в пакете UNSUBSCRIBE ДОЛЖНЫ быть закодированными строками UTF-8, как определено в пункте 4.5.3, формируя непрерывный пакет
[MQTT-6.10.4-1]	Фильтры тем (независимо от того, содержат ли они подстановочные знаки или нет), поставляемые в пакете UNSUBSCRIBE, ДОЛЖНЫ сравниваться по каждому символу с текущим набором фильтров тем, хранящихся на Сервере для Клиента. Если какой-либо фильтр точно соответствует, то его собственная Подписка удаляется, в противном случае никакой дополнительной обработки не происходит

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-6.10.4-2]	Если Сервер удаляет подписку, он ДОЛЖЕН завершить поставку любых сообщений QoS 1 или QoS 2, которые он начал отправлять Клиенту
[MQTT-6.10.4-3]	Если Сервер удаляет подписку, он ДОЛЖЕН прекратить добавлять новые сообщения для доставки Клиенту
[MQTT-6.10.4-4]	Сервер ДОЛЖЕН отвечать на запрос UNSUBSUBCRIBE, отправив пакет UNSUBACK. Пакет UNSUBACK ДОЛЖЕН иметь тот же идентификатор пакета, что и пакет UNSUBSCRIBE
[MQTT-6.10.4-5]	Даже если никакие Подписки на темы не удалены, Сервер ДОЛЖЕН отправить UNSUBACK
[MQTT-6.10.4-6]	Если Сервер получает пакет UNSUBSCRIBE, содержащий несколько фильтров тем, он ДОЛЖЕН обрабатывать этот пакет, как если бы он получил последовательность из нескольких пакетов UNSUBSCRIBE, за исключением того, что он отправляет только один ответ UNSUBACK
[MQTT-6.12.4-1]	Сервер ДОЛЖЕН отправить пакет PINGRESP в ответ на пакет PINGREQ
[MQTT-6.14.1-1]	Сервер ДОЛЖЕН проверить, чтобы зарезервированные биты были установлены на ноль, и прервать соединение с Клиентом, если они не равны нулю
[MQTT-6.14.4-1]	После отправки пакета DISCONNECT Клиент ДОЛЖЕН прервать сетевое соединение
[MQTT-6.14.4-2]	После отправки пакета DISCONNECT Клиент НЕ ДОЛЖЕН отправлять больше управляющих пакетов этим сетевым подключением
[MQTT-6.14.4-3]	При получении DISCONNECT Сервер ДОЛЖЕН отменить любое сообщение, связанное с текущим подключением, без его публикации, как описано в разделе 6.1.2.5
[MQTT-7.1.0-1]	Клиент и Сервер ДОЛЖНЫ сохранять состояние сеанса на протяжении всего сеанса
[MQTT-7.1.0-2]	Сеанс ДОЛЖЕН длиться как минимум до тех пор, пока есть активное сетевое соединение
[MQTT-7.3.1-1]	В протоколе доставки QoS 0 отправитель: ДОЛЖЕН отправить пакет PUBLISH с QoS = 0, DUP = 0
[MQTT-7.3.2-1]	В протоколе доставки QoS 1 отправитель ДОЛЖЕН: назначать неиспользуемый идентификатор пакета каждый раз, когда у него есть новое сообщение приложения для публикации; отправить пакет PUBLISH, содержащий этот идентификатор пакета с QoS = 1, DUP = 0; рассматривать пакет PUBLISH как «неподтвержденный», пока он не получит соответствующий пакет PUBACK от получателя. См. более подробную информацию о неподтвержденных сообщениях в подразделе 7.4.
[MQTT-7.3.2-2]	В протоколе доставки QoS 1 получатель ДОЛЖЕН: направить пакет PUBACK, содержащий идентификатор пакета из входящего пакета PUBLISH, приняв право собственности на сообщение приложения; После отправки пакета PUBACK считать любой входящий пакет PUBLISH, который содержит тот же идентификатор пакета, новой публикацией, независимо от установки его флага DUP

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-7.3.3-1]	В протоколе доставки QoS 2 отправитель: ДОЛЖЕН назначить неиспользуемый идентификатор пакета, если у него есть новое сообщение приложения для публикации; ДОЛЖЕН отправить пакет PUBLISH, содержащий этот идентификатор пакета, с QoS = 2, DUP = 0; ДОЛЖЕН считать пакет PUBLISH «неподтвержденным», пока он не получит соответствующий пакет PUBREC от получателя. См. информацию о неподтвержденных сообщениях в Разделе 4.4; ДОЛЖЕН отправить пакет PUBREL, когда он получает пакет PUBREC от получателя. Этот пакет PUBREL ДОЛЖЕН содержать тот же идентификатор пакета, что и исходный пакет PUBLISH; ДОЛЖЕН считать пакет PUBREL «неподтвержденным», пока он не получит соответствующий пакет PUBCOMP от получателя; НЕ МОЖЕТ повторно отправить PUBLISH после отправки соответствующего пакета PUBREL
[MQTT-7.3.3-2]	В протоколе доставки QoS 2 получатель ДОЛЖЕН: отвечать PUBREC, содержащим идентификатор пакета из входящего пакета PUBLISH, приняв право собственности на сообщение приложения; До тех пор, пока он не получит соответствующий пакет PUBREL, подтвердить любой последующий пакет PUBLISH с тем же идентификатором пакета, отправив PUBREC. В этом случае он НЕ ДОЛЖЕН инициировать дублирование сообщений для последующих получателей; В протоколе доставки QoS 2 получатель ДОЛЖЕН: отвечать PUBREC, содержащим идентификатор пакета из входящего пакета PUBLISH, приняв право собственности на сообщение приложения; До тех пор, пока он не получит соответствующий пакет PUBREL, подтвердить любой последующий пакет PUBLISH с тем же идентификатором пакета, отправив PUBREC. В этом случае он НЕ ДОЛЖЕН инициировать дублирование сообщений для последующих получателей; отвечать на пакет PUBREL, отправив пакет PUBCOMP, содержащий тот же идентификатор пакета, что и PUBREL; После того, как он отправил PUBCOMP, считать любой последующий PUBLISH-пакет, содержащий этот идентификатор пакета, новой публикацией
[MQTT-7.4.0-1]	Когда Клиент повторно подключается к параметру «Очистить сеанс», установленному на 0, Клиент и Сервер ДОЛЖНЫ повторно отправлять любые неподтвержденные пакеты PUBLISH (где QoS> 0) и пакеты PUBREL с использованием их исходных идентификаторов пакетов
[MQTT-7.5.0-1]	Когда Сервер получает право собственности на входящее сообщение приложения, он ДОЛЖЕН добавить его в состояние сеанса для тех Клиентов, которые имеют соответствующие Подписки. Правила соответствия определены в разделе 7.7
[MQTT-7.5.0-2]	Клиент ДОЛЖЕН подтвердить, что любой опубликованный пакет, который он получает в соответствии с применимыми правилами QoS, независимо от того, выбирает ли он обработку сообщения приложения о том, что оно содержит
[MQTT-7.6.0-1]	Когда он повторно отправляет какие-либо пакеты PUBLISH, он ДОЛЖЕН повторно отправить их в том порядке, в котором были отправлены исходные пакеты PUBLISH (это относится к сообщениям QoS 1 и QoS 2)
[MQTT-7.6.0-2]	Клиент ДОЛЖЕН отправлять пакеты PUBACK в том порядке, в котором были получены соответствующие пакеты PUBLISH (сообщения QoS 1)

Продолжение таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-7.6.0-3]	Клиент ДОЛЖЕН отправлять пакеты PUBREC в том порядке, в котором были получены соответствующие пакеты PUBLISH (сообщения QoS 2)
[MQTT-7.6.0-4]	Клиент ДОЛЖЕН отправлять пакеты PUBREL в том порядке, в котором были получены соответствующие пакеты PUBREC (сообщения QoS 2)
[MQTT-7.6.0-5]	Сервер ДОЛЖЕН по умолчанию рассматривать каждую тему как «упорядоченную тему». Он МОЖЕТ предоставить административный или иной механизм, позволяющий рассматривать одну или несколько тем как «неупорядоченную тему»
[MQTT-7.6.0-6]	Когда Сервер обрабатывает сообщение, опубликованное в упорядоченной теме, он ДОЛЖЕН следовать приведенным выше правилам при доставке сообщений каждому из своих подписчиков. Кроме того, он ДОЛЖЕН отправлять пакеты PUBLISH Клиентам (по той же теме и QoS) в том порядке, в котором они были получены от любого указанного Клиента
[MQTT-7.7.1-1]	Символы подстановки могут использоваться в фильтрах тем, но НЕ ДОЛЖНЫ использоваться в названии темы
[MQTT-7.7.1-2]	Многоуровневый подстановочный знак ДОЛЖЕН быть указан либо самостоятельно, либо после разделителя тематических уровней. В любом случае он ДОЛЖЕН быть последним символом, указанным в фильтре тем
[MQTT-7.7.1-3]	Одноуровневый шаблон можно использовать на любом уровне фильтра тем, включая первый и последний уровни. Если он используется, он ДОЛЖЕН занимать весь уровень фильтра
[MQTT-7.7.2-1]	Сервер НЕ ДОЛЖЕН соединять фильтры тем, которые начинаются с символа подстановки (# или +) с названиями тем, начинающимися с символа $
[MQTT-7.7.3-1]	Все названия тем и фильтры тем ДОЛЖНЫ иметь как минимум один символ
[MQTT-7.7.3-2]	Название темы и фильтры тем НЕ ДОЛЖНЫ включать нулевой символ (Unicode U + 0000)
[MQTT-7.7.3-3]	Название темы и фильтры темы — это кодированные строки UTF-8, они НЕ ДОЛЖНЫ кодировать более 65535 байт
[MQTT-7.7.3-4]	Когда выполняется соединение в соответствии с подпиской Сервер НЕ ДОЛЖЕН выполнять какую-либо стандартизацию названий тем или фильтров тем, а также любую модификацию или замену непризнанных символов
[MQTT-7.8.0-1]	Если не указано иное, если Сервер или Клиент сталкиваются с нарушением протокола, они ДОЛЖНЫ закрыть сетевое соединение, на которое получен такой управляющий пакет, который вызвал нарушение протокола
[MQTT-7.8.0-2]	Если Клиент или Сервер обнаруживают несистематическую ошибку при обработке входящего управляющего пакета, они ДОЛЖНЫ закрыть сетевое соединение, на которое получен такой управляющий пакет
[MQTT-8.0.0-1]	Управляющие пакеты MQTT ДОЛЖНЫ быть отправлены в бинарные пакеты данных WebSocket. Если получен какой-либо другой тип пакета данных, получатель ДОЛЖЕН прервать сетевое соединение
[MQTT-8.0.0-2]	Один пакет данных WebSocket может содержать множественные или частичные управляющие пакеты MQTT. Получатель НЕ ДОЛЖЕН предполагать, что управляющие пакеты MQTT соответствуют рамкам пакета данных WebSocket
[MQTT-8.0.0-3]	Клиент ДОЛЖЕН включать «mqtt» в список суб-протоколов WebSocket, который он предлагает
[MQTT-8.0.0-4]	Имя суб-протокола WebSocket, выбранное и возвращаемое Сервером, должно быть «mqtt»

Окончание таблицы Б.1

Номер нормативного положения	Нормативное положение
[MQTT-9.0.0-1]	Сервер, который принимает входящие подключения и устанавливает исходящие подключения к другим Серверам, ДОЛЖЕН соответствовать как Клиенту MQTT, так и Серверу MQTT
[MQTT-9.0.0-2]	Соответствующие реализации НЕ ДОЛЖНЫ требовать использования каких-либо расширений, определенных вне этой спецификации, чтобы взаимодействовать с любой другой совместимой реализацией
[MQTT-9.1.1-1]	Соответствующий Сервер ДОЛЖЕН поддерживать использование одного или нескольких базовых транспортных протоколов, которые обеспечивают упорядоченный, не допускающий потерь поток байтов от Клиента к Серверу и от Сервера к Клиенту
[MQTT-9.1.2-1]	Соответствующий Клиент ДОЛЖЕН поддерживать использование одного или нескольких базовых транспортных протоколов, которые обеспечивают упорядоченный, не допускающий потерь поток байтов от Клиента к Серверу и от Сервера к Клиенту

Приложение В
(справочное)

Предисловие к оригинальному изданию

ISO (Международная организация по стандартизации) и IEC (Международная электротехническая комиссия) образуют специализированную систему стандартизации во всем мире. Национальные органы, входящие в ISO или IEC, участвуют в разработке международных стандартов через технические комитеты, созданные соответствующей организацией для рассмотрения конкретных областей технической деятельности. Технические комитеты ISO и IEC сотрудничают в областях, представляющих взаимный интерес. В работе также принимают участие другие международные организации, правительственные и неправительственные, в сотрудничестве с ISO и IEC. В области информационных технологий ISO и IEC создали совместный *Технический комитет* ISO/IEC JTC 1.

Процедуры, используемые для разработки настоящего стандарта и предназначенные для его дальнейшего обслуживания, описаны в Директиве ISO/IEC, Часть 1. В частности, следует отметить различные критерии утверждения, необходимые для разных типов документов. Настоящий стандарт составлен в соответствии с редакционными правилами Директивы ISO/IEC, Часть 2 (см. gost.e-ecolog.ru).

Следует обратить внимание на возможность того, что некоторые элементы настоящего стандарта могут быть предметом патентных прав. ISO и IEC не несут ответственности за выявление любых или всех таких патентных прав. Подробная информация о каких-либо патентных правах, определенных при разработке стандарта, будет представлена во введении и/или в списке заявок на патент ISO (см. gost.e-ecolog.ru .

Любое торговое название, используемое в настоящем стандарте, является информацией, предоставленной для удобства пользователей, и не является подтверждающим качество.

ISO/IEC 20922:2016 был подготовлен *Техническим комитетом* по Сетевому протоколу обмена сообщениями между устройствами (MQTT) в OASIS и был принят в соответствии с процедурой PAS Совместным техническим комитетом ISO/IEC JTC 1 «Информационные технологии» параллельно с его утверждением национальными органами ISO и IEC.

Приложение Г
(справочное)

Извещение правообладателя оригинального издания

Авторское право © OASIS Open 2014. Все права защищены.

Все термины с заглавной буквы в следующем тексте имеют значения, присвоенные им в Политике OASIS по защите прав на интеллектуальную собственность («Политика OASIS по ЗАПИСИ»). Полный текст политики можно найти на веб-сайте OASIS.

Настоящий стандарт и его переводы могут быть скопированы и предоставлены сторонним лицам, а производные работы, которые комментируют или иным образом объясняют его или помогают в его осуществлении, могут быть подготовлены, скопированы, опубликованы и распространены полностью или частично без каких-либо ограничений при условии, что вышеупомянутое уведомление об авторских правах и данный раздел включены во все такие копии и производные работы. Тем не менее, настоящий стандарт сам по себе не может быть каким-либо образом изменен, в том числе путем удаления уведомления об авторских правах или ссылок на OASIS, за исключением случаев, когда это необходимо для разработки любого документа или проектной документации Техническим комитетом OASIS (в этом случае должны соблюдаться правила, применимые к авторским правам, как указано в Политике OASIS по ЗПИС) или, если требуется выполнить перевод на другие языки, кроме английского.

Ограниченные разрешения, предоставленные выше, являются бессрочными и не будут отозваны OASIS, его преемниками или назначенными лицами.

Настоящий стандарт и содержащаяся в нем информация предоставляются «в существующем виде», и OASIS ОТКАЗЫВАЕТСЯ ОТ ВСЕХ ГАРАНТИЙ, ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ЛЮБЫМИ ГАРАНТИЯМИ, ЧТО ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИИ, СОДЕРЖАЩЕЙСЯ В НАСТОЯЩЕМ СТАНДАРТЕ, НЕ НАРУШАЕТ КАКИЕ-ЛИБО ПРАВА СОБСТВЕННОСТИ ИЛИ ЛЮБЫЕ ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ КОММЕРЧЕСКОЙ ЦЕННОСТИ ИЛИ ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

OASIS просит, чтобы какая-либо Сторона OASIS или любая другая сторона, которая считает, что она имеет патентные претензии, которые непременно будут нарушены в результате применения настоящего стандарта Комитета OASIS или Стандарта OASIS, уведомила Администратора ТК OASIS и сообщила о готовности предоставить патентные лицензии по таким патентным претензиям в соответствии с режимом ЗПИС Технического комитета OASIS, который подготовил настоящую Спецификацию.

OASIS предлагает любой из сторон связаться с Администратором ТК OASIS, если ему известно о требовании права собственности на любые патентные претензии, которые непременно будут нарушены в результате реализации настоящей Спецификации владельцем патента, который не желает предоставлять лицензию на такие патентные претензии в соответствии с режимом ЗПИС Технического комитета OASIS, который подготовил настоящую Спецификацию. OASIS может включать такие претензии на своем веб-сайте, но отказывается от каких-либо обязательств в этом отношении.

OASIS не занимает позицию относительно действительности или сферы применения любой интеллектуальной собственности или других прав, которые могут быть заявлены в отношении реализации или использования технологии, описанной в настоящем стандарте, или в степени, в которой любая лицензия по таким правам может быть или не быть доступной; комитет также не заявляет о том, что приложил все усилия для выявления таких прав. Информацию о процедурах OASIS в отношении прав на любой документ или проектную документацию, подготовленную Техническим комитетом OASIS, можно найти на веб-сайте OASIS. Копии заявлений о правах, предоставленных для публикации, и любые гарантии выдачи лицензий, или результат попытки получить общую лицензию или разрешение на использование таких имущественных прав исполнителями или пользователями настоящей Спецификации Комитета OASIS или Стандарт OASIS, можно получить у Администратора ТК OASIS. OASIS не делает никаких заявлений о том, что любая информация или список прав интеллектуальной собственности в любой момент будет завершен, или что любые претензии в таком списке фактически являются Основными претензиями.

Название «OASIS» является товарным знаком OASIS, владельца и разработчика настоящей спецификации, и его следует использовать только для обозначения организации и ее официальных продуктов. OASIS приветствует упоминание, применение и использование спецификаций, сохраняя право принудительного применения своих знаков в отношении вводящих в заблуждение видов использования. Вышеуказанные рекомендации см. по адресу:

gost.e-ecolog.ru

Приложение Д
(справочное)

Сведения об изменениях, внесенных в текст национального стандарта по отношению к исходному стандарту

Стандарт «Информационные технологии. Интернет вещей. Протокол организации очередей доставки телеметрических сообщений MQTT. Версия 3.1.1» является модифицированным по отношению к международному стандарту ИСО/МЭК 20922:2016 Информационные технологии. Протокол организации очередей доставки телеметрических сообщений MQTT v3.1.1 (ISO/IEC 20922:2016 Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1) путем изменения его структуры в соответствии с условиями, указанными в п В.13 ГОСТ 1.3—2014 с целью приведения его в соответствие требованиям, установленным в ГОСТ 1.5.

В процессе модификации в исходный стандарт были внесены следующие изменения:

а) Добавление разделов к исходному стандарту

1 Добавлены разделы «Предисловие», «Введение», «Область применения», «Обозначения и сокращения», «Библиография».

2 Текст раздела «Область применения» добавлен на основе положений исходного стандарта.

3 В раздел «Термины и определения» добавлены основные положения документа [1] как значимые для изложения национального стандарта. Вставленный текст выделен вертикальной линией. Добавлен термин 2.11, выделен курсивом.

б) Удаление разделов исходного стандарта

1 Убраны подразделы исходного стандарта «URL-спецификации», «Технический комитет», «Председатели», «Редакторы», «Смежные направления исследований», «Формат ссылки», «Список рисунков и таблиц», «1.1 Организация MQTT», «История редакций», “Выражение признательности”.

в) Перенос разделов исходного стандарта.

1 Текст раздела «Предисловие» исходного стандарта перенесен в приложение В (справочное) национального стандарта.

2 Текст подраздела «Выдержка» исходного стандарта перенесен в раздел «Введение» национального стандарта.

3 Текст подраздела «Извещения» исходного стандарта перенесен в приложение Г (справочное) национального стандарта.

4 Текст подраздела «1.3 Ссылки на нормативные документы» исходного стандарта перенесен в раздел «Библиография» национального стандарта, при этом ссылочные знаки и нумерация приведены в соответствие с российскими требованиями по стандартизации.

5 Положения подраздела «1.6 Редакционные соглашения» исходного стандарта включены в текст раздела «Область применения» национального стандарта.

6 Раздел «5 Безопасность» исходного стандарта перенесен в приложение А (рекомендуемое) национального стандарта.

7 Изменен порядок следования приложений национального стандарта с учетом добавления новых приложений.

г) Изменения текста исходного стандарта в соответствии с требованиями Российской стандартизации.

1 Изменена нумерация заголовков 1-го уровня и приложений исходного стандарта.

2 Рисунки представлены в виде таблиц. Нумерация таблиц в соответствии с нумерацией разделов отменена. Таблицам присвоена сплошная нумерация по нарастанию по всему тексту стандарта.

3 Изменена нумерация обозначений обязательных нормативных положений, предусмотренных п. 3.1.1 национального стандарта, а также внутренних ссылок на разделы в соответствии с изменившейся нумерацией разделов.

4 Из текста исходного стандарта исключено выделение цветом.

5 Из текста исходного стандарта исключены словосочетания «Ненормативный пример» и «Ненормативный комментарий», соответствующие блоки текста выделены как примеры или как примечания (в зависимости от содержания соответствующего блока) в соответствии с требованиями ГОСТ 1.5.

Библиография

[1] Брэднер С., «Ключевые слова, используемые в RFC для указания уровня требований», BCP 14, RFC 2119, март 1997 г. (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 58 1997.) gost.e-ecolog.ru

[2] Ержо Ф., «UTF-8, формат преобразования ISO 10646», STD 63, RFC 3629, ноябрь 2003 г. (Yergeau, F., “UTF-8, a transformation format of ISO 10646”, STD 63, RFC 3629, November 2003) gost.e-ecolog.ru

[3] Диркс, Т. и Е. Рескорла, «Протокол безопасности транспортного уровня (TLS), версия 1.2», RFC 5246, август 2008 г. (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, RFC 5246, August 67 2008.) gost.e-ecolog.ru

[4] Фет, И. и А. Мельников, «Протокол WebSocket», RFC 6455, декабрь 2011 г. gost.e-ecolog.ru

[5] Консорциум Unicode. Стандарт Unicode. gost.e-ecolog.ru

[6] Постель Дж., Протокол управления передачей. STD 7, IETF RFC 793, сентябрь 1981 г. (Postel, J. Transmission Control Protocol. STD 7, IETF RFC 793, September 1981) gost.e-ecolog.ru

[7] Передовой стандарт шифрования (AES) (FIPS PUB 197). (Advanced Encryption Standard (AES) (FIPS PUB 197)) gost.e-ecolog.ru

[8] Стандарт шифрования данных (DES). (Data Encryption Standard (DES)) gost.e-ecolog.ru fips46-3/fips46-3.pdf

[9] Требования безопасности для криптографических модулей (FIPS PUB 140-2) (Security Requirements for Cryptographic Modules (FIPS PUB 140-2)) gost.e-ecolog.ru

[10] Стандарт IEEE для локальных и городских сетей — идентификация безопасного устройства (IEEE Standard for Local and metropolitan area networks — Secure Device Identity) gost.e-ecolog.ru standard/802.1AR-2009.html

[11] ISO/IEC 29192-1:2012 Информационные технологии. Методы безопасности. Легкая криптография. Часть 1. Общая информация (ISO/IEC 29192-1:2012 Information technology — Security techniques — Lightweight cryptography — Part 1: General) gost.e-ecolog.ru

[12] Дополнительная публикация MQTT, структура MQTT и NIST для улучшения ключевой инфраструктуры кибербезопасности (MQTT supplemental publication, MQTT and the NIST Framework for Improving Critical Infrastructure Cybersecurity) gost.e-ecolog.ru

[13] Постановление Правительства США 13636 об усовершенствовании ключевой инфраструктуры кибербезопасности (Improving Critical Infrastructure Cybersecurity Executive Order 13636) gost.e-ecolog.ru preliminary-cybersecurity-framework.pdf

[14] NISTIR 7628 Руководство по безопасности интеллектуальных сетей (NISTIR 7628 Guidelines for Smart Grid Cyber Security) gost.e-ecolog.ru

[15] Криптография NSA № B (NSA Suite B Cryptography) gost.e-ecolog.ru

[16] Стандарт безопасности данных PCI-DSS для платежных карт (PCI-DSS Payment Card Industry Data Security Standard) gost.e-ecolog.ru

[17] Лич М., Генис М., Ли У., Керис Р., Коблас Д. и Л. Джонс, «Протокол SOCKS 5», RFC 1928, март 1996 г. (Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and L. Jones, “SOCKS Protocol Version 5”, RFC 1928, March 1996.) http:// gost.e-ecolog.ru 28.txt

[18] Семершайм Дж., ред., «Протокол доступа к легким протоколам (LDAP): протокол», RFC 4511, июнь 2006 г. (Sermersheim, J., Ed., «Lightweight Directory Access Protocol (LDAP): The Protocol», RFC 4511, June 2006.) http:// gost.e-ecolog.ru .txt

[19] Сэлоуэй Дж., Чу Г., Эронен П. и Г. Тшофениг, «Возобновление сеанса безопасности транспортного уровня (TLS) без состояния на стороне сервера», RFC 5077, январь 2008 г. (Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, «Transport Layer Security (TLS) Session Resumption without Server-Side State», RFC 5077, January 2008) gost.e-ecolog.ru

[20] Купер Д., Сантессон С., Фарелл С., Боэйен С., Хаусли Р. и В. Польк, «Internet X.509 Сертификат инфраструктуры открытых ключей и список отзыва сертификатов (CRL)», RFC 5280, май 2008 г. (Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, «Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile», RFC 5280, May 2008) gost.e-ecolog.ru

[21] Истлейк 3 Д., «Расширение безопасности транспортного уровня (TLS): определения расширений», RFC 6066, январь 2011 г. (Eastlake 3rd, D., «Transport Layer Security (TLS) Extensions: Extension Definitions», RFC 6066, January 2011) gost.e-ecolog.ru

[22] Хадт Д., ред., «Инфраструктура авторизации Auth 2.0», RFC 6749, октябрь 2012 г. (Hardt, D., Ed., «The OAuth 2.0 Authorization Framework», RFC 6749, October 2012) gost.e-ecolog.ru

[23] Сантессон С., Майерс М., Эенкни Р., Малпани А., Галперин С. И К. Адамс «X.509 Интернет-протокол проверки подлинности в Интернете с открытым ключом — OCSP», RFC 6960, июнь 2013. (Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol — OCSP", RFC 6960, June 2013) gost.e-ecolog.ru

[24] Закон Сарбейнса-Оксли 2002 г. (Sarbanes-Oxley Act of 2002) gost.e-ecolog.ru html/PLAW-107publ204.htm

[25] Зона безопасности США-ЕС (U.S.-EU Safe Harbor) gost.e-ecolog.ru

УДК 004:006.354 ОКС 35.100.70

Ключевые слова: информационные технологии, интернет вещей, протокол MQTT

БЗ 9—2019/14

Редактор *Г.Н. Симонова*
Технический редактор *И.Е. Черепкова*
Корректор *И.А. Королева*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 21.10.2019. Подписано в печать 28.10.2019. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 6,98. Уч.-изд. л. 6,28.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении во 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