

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра суперкомпьютеров и общей информатики

Отчет по лабораторной работе №2

Дисциплина: «Технологии Интернета вещей»

Выполнил: Мелешенко И.С.

Группа: 6133-010402D.

Дата: 16.05.2022

Самара 2022

СОДЕРЖАНИЕ

1. Место протокола MQTT в стеке протоколов TCP/IP	4
2. Какие протоколы использует MQTT для передачи информации на транспортном сетевом, канальном и физическом уровнях.	5
3. Описание (общие черты) протокола MQTT	6
4. Опишите принцип "Издатель- подписчик" и как он работает в MQTT. Приведите пример.	7
5. Что такое брокер? Как он работает.	8
6.Опишите формат сообщения протокола MQTT	9
7. Назначение сообщения CONNECT. опишите основные поля сообщения и их назначение.....	10
8. Назначение сообщения CONNACK. опишите основные поля сообщения и их назначение.....	12
9.Назначение сообщения SUBSCRIBE. опишите основные поля сообщения и их назначение.....	13
10.Назначение сообщения SUBACK. опишите основные поля сообщения и их назначение.....	14
11.Назначение сообщения PUBLISH. опишите основные поля сообщения и их назначение.....	15
12.Назначение сообщения PUBACK. опишите основные поля сообщения и их назначение.....	17
13.Назначение сообщения PUBREC. опишите основные поля сообщения и их назначение.....	18
14.Назначение сообщения PUBREL. опишите основные поля сообщения и их назначение.....	19
15.Назначение сообщения PUBCOMP. опишите основные поля сообщения и их назначение.....	20
16.Назначение сообщения UNSUBSCRIBE. опишите основные поля сообщения и их назначение.	21

17.Назначение сообщения UNSUBACK. опишите основные поля сообщения и их назначение.....	22
18.Назначение сообщения PINGREQ. опишите основные поля сообщения и их назначение.....	23
19.Назначение сообщения PINGRESP. опишите основные поля сообщения и их назначение.....	24
20.Назначение сообщения DISCONNECT. опишите основные поля сообщения и их назначение.	25
21. Уровни QoS в MQTT. Примеры использования уровней.....	26
22. Семантика тем в MQTT. одноуровневая тема, многоуровневая тема. Пример темы. Как обеспечивается получение данных по нескольким темам.	29
23. Проблемы MQTT.....	31

1. МЕСТО ПРОТОКОЛА MQTT В СТЕКЕ ПРОТОКОЛОВ TCP/IP

Протокол MQTT – Message Queuing Telemetry Transport – протокол для передачи последовательности сообщений с телеметрическими данными, то есть информации от датчиков температуры, влажности, освещенности и др.

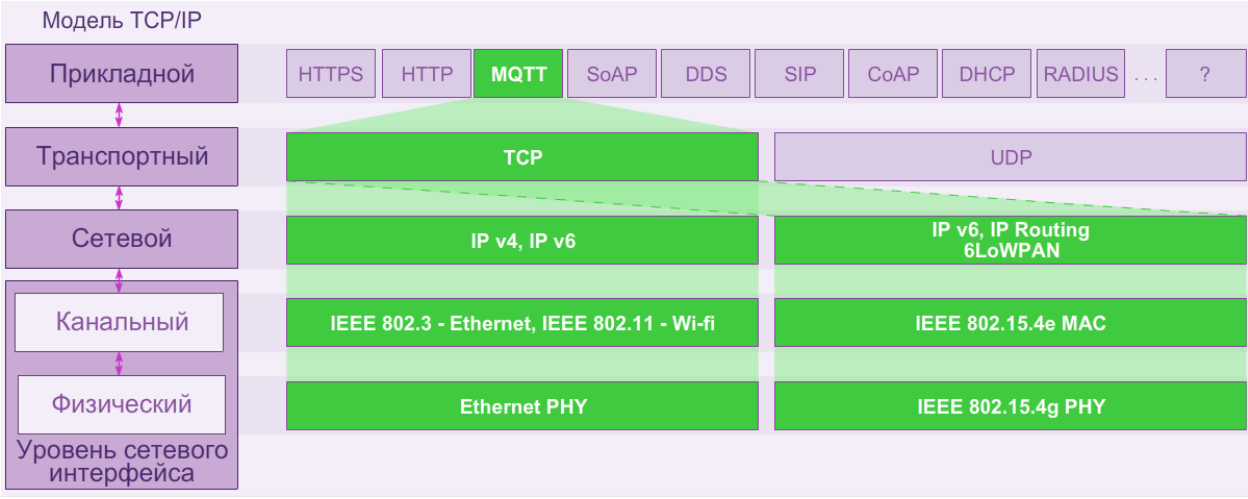
MQTT был предложен в 1999 г. Энди Стандфордом-Кларком в качестве протокола, который бы служил для передачи данных о состоянии нефтепровода и газопровода в реальном времени. Разработка велась компанией IBM для нового трубопровода крупнейшей американской нефтяной компании ConocoPhillips.

Впервые протокол MQTT был опубликован консорциумом OASIS (Organization for the Advancement of Structured Information Standards) в октябре 2014 г.

В июне 2016 г. стандарт был признан Международной организацией по стандартизации (ISO). MQTT Version 3.1.1 был зарегистрирован техническим комитетом по информационным технологиям ISO (JTC1) под номером ISO/IEC 20922.

Протокол создавался, чтобы обеспечить открытость, простоту, минимальные требования к ресурсам и удобство внедрения. MQTT – бинарный протокол обмена сообщениями, работает на **прикладном уровне поверх TCP/IP** и использует по умолчанию 1883 порт (8883 при подключении через SSL).

2. КАКИЕ ПРОТОКОЛЫ ИСПОЛЬЗУЕТ MQTT ДЛЯ ПЕРЕДАЧИ ИНФОРМАЦИИ НА ТРАНСПОРТНОМ СЕТЕВОМ, КАНАЛЬНОМ И ФИЗИЧЕСКОМ УРОВНЯХ.



3. ОПИСАНИЕ (ОБЩИЕ ЧЕРТЫ) ПРОТОКОЛА MQTT

Основные черты протокола MQTT:

- обмен сообщениями происходит по принципу "издатель-подписчик" (Pub-Sub);
- размер заголовка сообщения составляет 2 байта, а полезная нагрузка может варьироваться от 1 байта до 260 Мбайт;
- в протоколе заложена возможность выбора одного из трех уровней обслуживания.

4. ОПИШИТЕ ПРИНЦИП "ИЗДАТЕЛЬ- ПОДПИСЧИК" И КАК ОН РАБОТАЕТ В MQTT. ПРИВЕДИТЕ ПРИМЕР.

В сети на базе протокола MQTT различают три объекта:

- Издатель (Publisher) – MQTT-клиент, который при возникновении определенного события передает брокеру информацию о нем, публикуя соответствующие темы (Topic);
- Брокер (Broker) – MQTT-сервер, который принимает информацию от издателей и передает ее соответствующим подписчикам, в сложных системах может выполнять также различные операции, связанные с анализом и обработкой поступивших данных. Разные брокеры могут соединяться между собой, если они подписываются на сообщения друг друга;
- Подписчик (Subscriber) – MQTT-клиент, который после подписки к брокеру большую часть времени «слушает» его и постоянно готов к приему и обработке входящего сообщения на интересующие темы (Topic) от брокера.

Протокол MQTT реализует принцип «издатель-подписчик».

Отличительной особенностью принципа «издатель-подписчик» от клиент-серверного подхода является то, что клиенты, посылающие сообщения (издатели, Publisher), и клиенты, принимающие сообщения (подписчики, Subscriber), как правило, разделены.

Разделение может быть организовано в трех плоскостях:

- пространство – издатель и подписчик не обязаны знать друг друга;
- время – издатель и подписчик не должны быть включены в одно и то же время;
- синхронизация – операции на обеих сторонах не должны приостанавливаться в течение публикации или получения информации.

5. ЧТО ТАКОЕ БРОКЕР? КАК ОН РАБОТАЕТ.

Брокер представляет собой некое средство синхронизации издателей и подписчиков. Брокер получает сообщения от издателей, записывает сообщения в определенные топики, а затем эти сообщения уходят к подписчикам, которые на эти топики подписаны.

6.ОПИШИТЕ ФОРМАТ СООБЩЕНИЯ ПРОТОКОЛА MQTT

Сообщение протокола MQTT состоит из :

- Fixed header, Заголовок фиксированной длины (присутствует во всех сообщениях). Первый байт заголовка фиксированной длины включает четыре поля, три из которых являются специальными флагами Flags (DUP, QoS Level, RETAIN), наличие и состояние которых зависит от типа сообщения, а четвертое указывает тип сообщения - MQTT Control Packet type.

Второй байт - Remaining Length - служит для указания оставшейся длины сообщения, которая складывается из размера заголовка переменной длины (если он есть) и размера полезной нагрузки, может занимать от 1 до 4 байт.

- Variable header, Заголовок переменной длины (присутствует только в определенных сообщениях). Variable header

Некоторые сообщения протокола MQTT содержат заголовок переменной длины, который располагается между заголовком фиксированной длины (Fixed header) и полем полезной нагрузки Payload. Наличие заголовка переменной длины и его содержание зависит от типа сообщения.

- Payload, Поля полезной нагрузки переменной длины (присутствует не во всех сообщениях). Некоторые сообщения протокола MQTT содержат полезную нагрузку. Содержание и формат данных, передаваемых в поле Payload сообщения MQTT, зависит от типа приложения. Размер передаваемых данных может быть вычислен путём вычитания из значения поля Remaining Length длины переменного заголовка.

Рассмотрим более подробно формат каждого сообщения MQTT.

7. НАЗНАЧЕНИЕ СООБЩЕНИЯ CONNECT. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

С помощью сообщения CONNECT клиент запрашивает подключение к серверу. Первое сообщение, отправленное от клиента серверу должно быть сообщением CONNECT.



Сообщение CONNECT содержит фиксированный заголовок, заголовок переменной длины и полезную нагрузку. Поле Remaining Length содержит значение оставшейся длины сообщения (длина заголовка переменной длины (10 байт) + длина поля полезной нагрузки Payload).

Заголовок переменной длины сообщения CONNECT состоит из четырех полей: Protocol Name, Protocol Level, Connect Flags, and Keep Alive.

Поле Protocol Name присутствует в заголовке переменной длины только в сообщениях типа CONNECT. Поле Protocol Name содержит название протокола «MQTT», закодированной в формате UTF-8. Длина поля Protocol Name составляет 6 байт. Поле Protocol Level присутствует в заголовке переменной длины только в сообщениях типа CONNECT. Поле Protocol Level содержит редакцию протокола, используемую клиентом. Значение поля

Protocol Level для версии MQTT 3.1.1 равно 4. Поле Connect Flags заголовка переменной длины содержит ряд параметров, определяющих поведение клиента при MQTT соединении. Оно также указывает на наличие или отсутствие определенных полей в полезной нагрузке.

Поле Keep Alive присутствует в заголовке переменной длины сообщения CONNECT. Значение поля Keep Alive, измеряемое в секундах, определяет максимальный временной интервал между сообщениями, получаемыми от клиента, что позволяет серверу обнаружить обрыв соединения с клиентом без ожидания таймаута по TCP/IP.

Клиент обязан посылать сообщение в течение каждого периода, указанного в поле Keep Alive. В случае отсутствия сообщения с передаваемыми данными в течение этого периода времени, клиент отправляет сообщение PINGREQ, которое сервер подтверждает сообщением PINGRESP. Если сервер не получает сообщение от клиента в течение полуторного периода, указанного в поле Keep Alive, он разъединяется с клиентом, как если бы клиент отправил сообщение DISCONNECT. Это действие не влияет ни на одну из подписок клиента.

Поле Keep Alive длиной 16 бит содержит значение временного интервала в секундах. Фактическое значение интервала времени зависит от приложения, но чаще всего составляет несколько минут. Максимальное значение составляет около 18 часов. Значение ноль (0) означает, что клиент не отключен.

Поле полезной нагрузки состоит из одного и более закодированных полей. Они содержат уникальный идентификатор клиента (Client Identifier), Will topic, Will Message, имя пользователя и пароль. Все данные кроме Client identifier является опциональной информацией и их присутствие определяется на основе флагов в заголовке переменной длины.

8. НАЗНАЧЕНИЕ СООБЩЕНИЯ CONNACK. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

CONNACK – Acknowledge Connection Request

Сообщение CONNACK отправляется сервером в ответ на сообщение CONNECT, полученное от клиента. Первым сообщением, отправленным от Сервера Клиенту, должно быть сообщение CONNACK.

Если клиент не получает от сервера сообщение CONNACK в течение определенного времени, то он отключается от сервера. Величина времени ожидания зависит от типа приложения и сетевой инфраструктуры.

Сообщение CONNACK содержит заголовок фиксированной длины и заголовок переменной длины. Полезная нагрузка Payload в сообщении CONNACK отсутствует.

Заголовок переменной длины содержит поле Connect Acknowledge Flags и Connect Return Code.

9. НАЗНАЧЕНИЕ СООБЩЕНИЯ SUBSCRIBE. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

Сообщение SUBSCRIBE передается от клиента к серверу для создания одной или нескольких подписок. Каждая подписка регистрирует интерес клиента по одной или нескольким темам. В сообщении SUBSCRIBE также указывается (для каждой подписки) максимальная величина QoS, с которой сервер может посылать сообщения приложения клиенту.

Сообщение SUBSCRIBE содержит заголовок фиксированной длины, заголовок переменной длины и полезную нагрузку.

Биты 3,2,1 и 0 заголовка фиксированной длины сообщения SUBSCRIBE зарезервированы и должны быть установлены - 0,0,1 и 0 соответственно. Сервер должен рассматривать любое другое значение как неправильное и закрыть сетевое подключение.

Заголовок переменной длины содержит поле Packet Identifier - идентификатор пакета.

Полезная нагрузка сообщения SUBSCRIBE содержит список фильтров тем, указывающих темы, на которые клиент хочет подписаться. Сервер должен поддерживать фильтры тем, содержащие символы wildcard. За каждым фильтром следует байт, называемый запрашиваемый QoS. Что позволяет запросить максимальный уровень качества обслуживания, при котором сервер может отправлять сообщения приложения клиенту. Полезная нагрузка сообщения SUBSCRIBE должна содержать как минимум одну пару фильтр темы/качество обслуживания QoS. Сообщение SUBSCRIBE без полезной нагрузки является некорректным.

10. НАЗНАЧЕНИЕ СООБЩЕНИЯ SUBACK. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

SUBACK – Subscribe acknowledgement

Сообщение SUBACK отправляется Сервером Клиенту для подтверждения получения и обработки сообщения SUBSCRIBE.

Сообщение SUBACK содержит список кодов возврата, указывающих максимальный уровень качества обслуживания, который гарантируется для каждой подписки, запрошенной в сообщении SUBSCRIBE.

Сообщение SUBSCRIBE содержит заголовок фиксированной длины, заголовок переменной длины и полезную нагрузку.

Заголовок переменной длины содержит поле Packet Identifier, значение которого равно значению поля Packet Identifier сообщения SUBSCRIBE, на которое передается подтверждение.

Полезная нагрузка содержит список кодов возврата. Каждый код возврата соответствует фильтру темы в подтверждаемом сообщении SUBSCRIBE.

11. НАЗНАЧЕНИЕ СООБЩЕНИЯ PUBLISH. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

Сообщение PUBLISH отправляется от клиента к серверу или от сервера к клиенту для публикации сообщения приложения.

Сообщение PUBLISH состоит из заголовка фиксированной длины, заголовка переменной длины и полезной нагрузки.

Поле Flags

Четыре старших бита первого байта фиксированного заголовка отведены под специальные флаги:

- DUP (Duplicate delivery)

1 байт, 3-й бит

Специальный флаг DUP – дублирование сообщения. Этот флаг указывает получателю, что полученное им сообщение передается повторно и, возможно, уже было получено им ранее. Этот флаг играет важную роль при передаче информации по ненадежным каналам, где возможна потеря сигнала.

- QoS level

1 байт, 2-1 биты

Этот флаг указывает уровень надежности доставки сообщения PUBLISH.

- RETAIN

Байт 1, бит 0

Этот флаг используется только для сообщения PUBLISH. Когда клиент отправляет сообщения PUBLISH серверу, если флаг RETAIN установлен в 1, сервер должен сохранить сообщение после доставки соответствующим подписчикам.

Когда новый подписчик присоединяется к теме, то последнее сохраненное сообщение по этой теме должно быть отправлено новому подписчику с установленным флагом RETAIN. В случае если сохраненные сообщения отсутствуют, то новому подписчику ничего не отправляется.

Заголовок переменной длины содержит поля Topic Name и Packet Identifier.

Поле Topic Name передается в заголовке переменной длины сообщения PUBLISH и указывает название темы, данные по которой содержатся в поле полезной нагрузки для публикации.

Payload

Полезная нагрузка содержит сообщение приложения для публикации. Содержание и формат данных зависят от приложения. Длина полезной нагрузки может быть вычислена путем вычитания длины заголовка переменной из значения поля Remaining Length, которое находится в заголовке фиксированной длины. Сообщение PUBLISH также может содержать нулевую длину полезной нагрузки.

Получатель сообщения PUBLISH в зависимости от значения флага QoS Level должен ответить следующим образом:

- Получение сообщения PUBLISH с QoS level 0 не требует ответа от получателя.

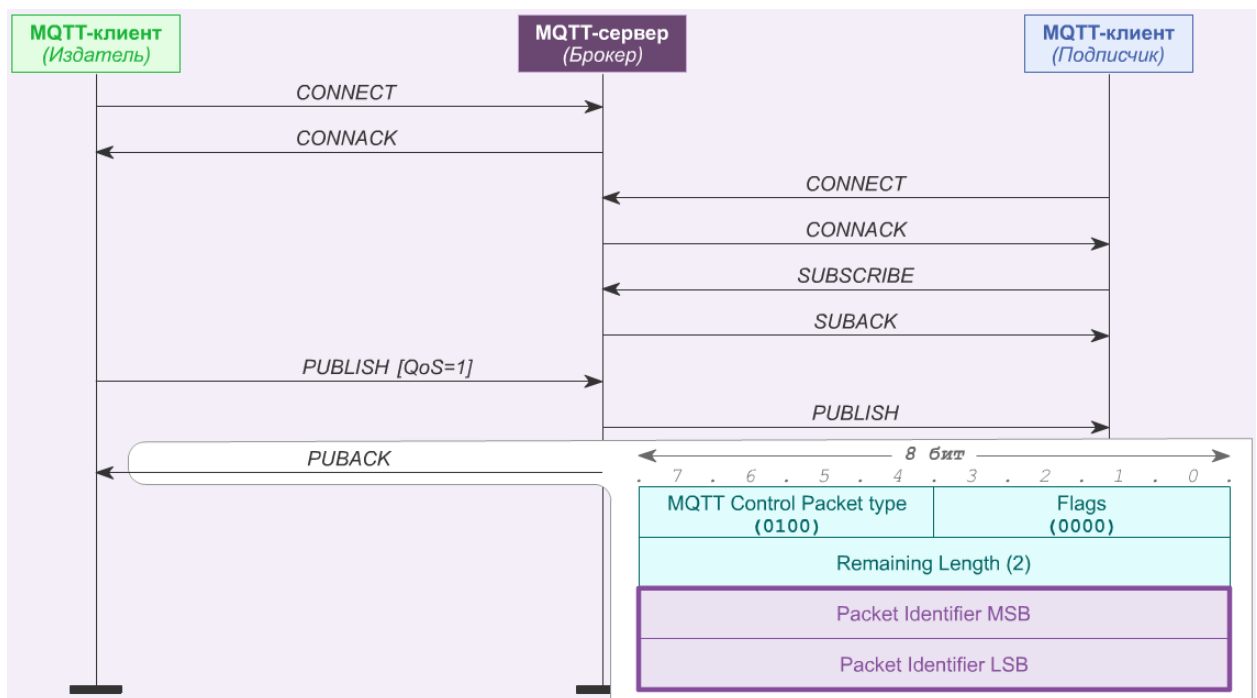
- В ответ на сообщение PUBLISH с QoS level 1 получатель должен отправить сообщение PUBACK.

- В ответ на сообщение PUBLISH с QoS level 2 получатель должен отправить сообщение PUBREC.

12. НАЗНАЧЕНИЕ СООБЩЕНИЯ PUBACK. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

Сообщение PUBACK-это ответ на сообщение PUBLISH с QoS=1. Сообщение PUBACK содержит заголовок фиксированной длины и заголовок переменной длины. Полезная нагрузка в сообщении PUBACK отсутствует.

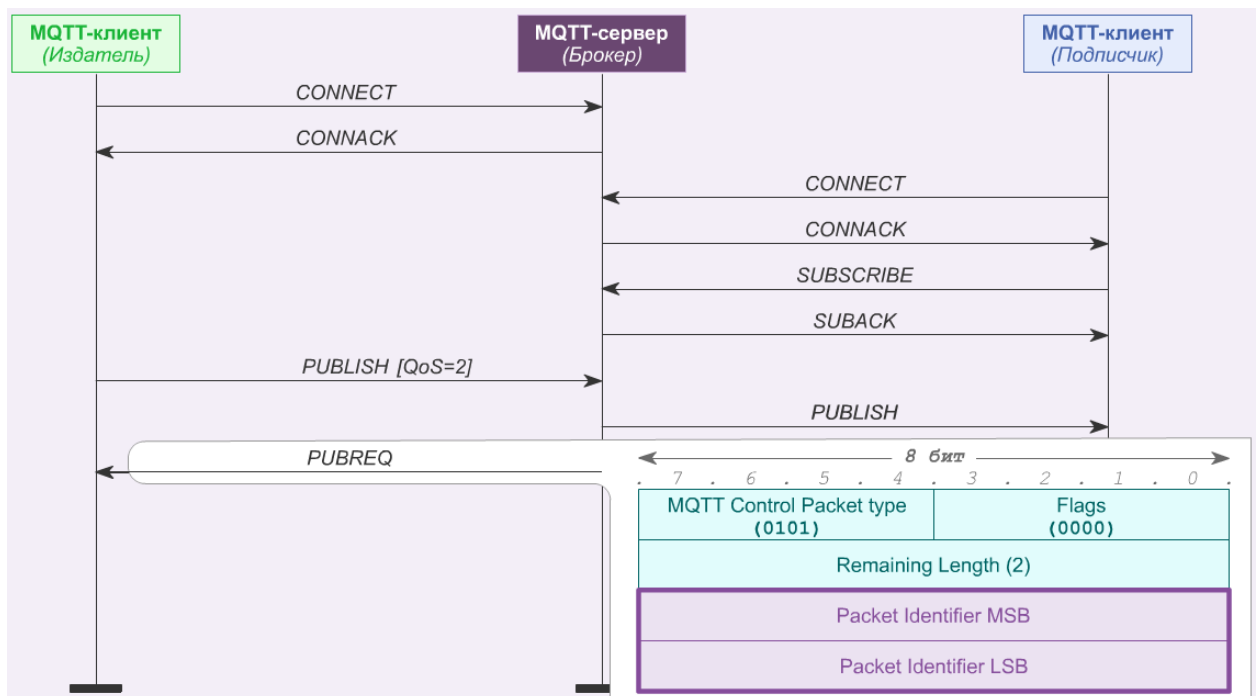
Заголовок переменной длины содержит Packet Identifier - идентификатор пакета из сообщения PUBLISH, на которое отправляется подтверждение.



13. НАЗНАЧЕНИЕ СООБЩЕНИЯ PUBREC. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

Сообщение PUBREC – это ответ на сообщение PUBLISH с QoS =2. Это второе сообщение в сценарии обмена сообщениями публикаций с QoS=2. Сообщение PUBREC содержит заголовок фиксированной длины и заголовок переменной длины. Полезная нагрузка в сообщении PUBREC отсутствует.

Заголовок переменной длины содержит Packet Identifier из сообщения PUBLISH, на которое отправляется подтверждение.



14. НАЗНАЧЕНИЕ СООБЩЕНИЯ PUBREL. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

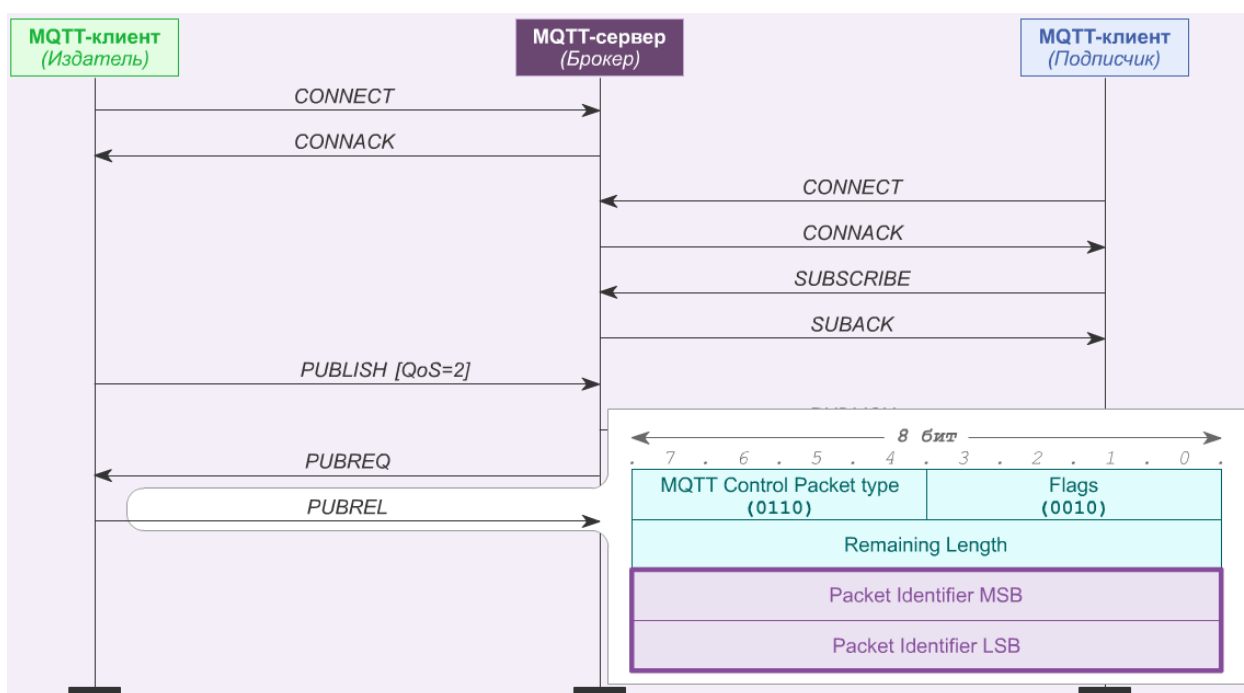
Сообщение PUBREL – это ответ на сообщение PUBREC. Это третье сообщение в сценарии обмена сообщениями публикаций с QoS=2.

Сообщение PUBREL содержит заголовок фиксированной длины и заголовок переменной длины. Полезная нагрузка в сообщение PUBREL отсутствует.

Поле Flags

Биты 3,2,1 и 0 фиксированного заголовка в сообщении PUBREL зарезервированы и должны быть установлены на 0,0,1 и 0 соответственно. Сервер должен рассматривать любое другое значение, как неправильное и закрыть сетевое подключение.

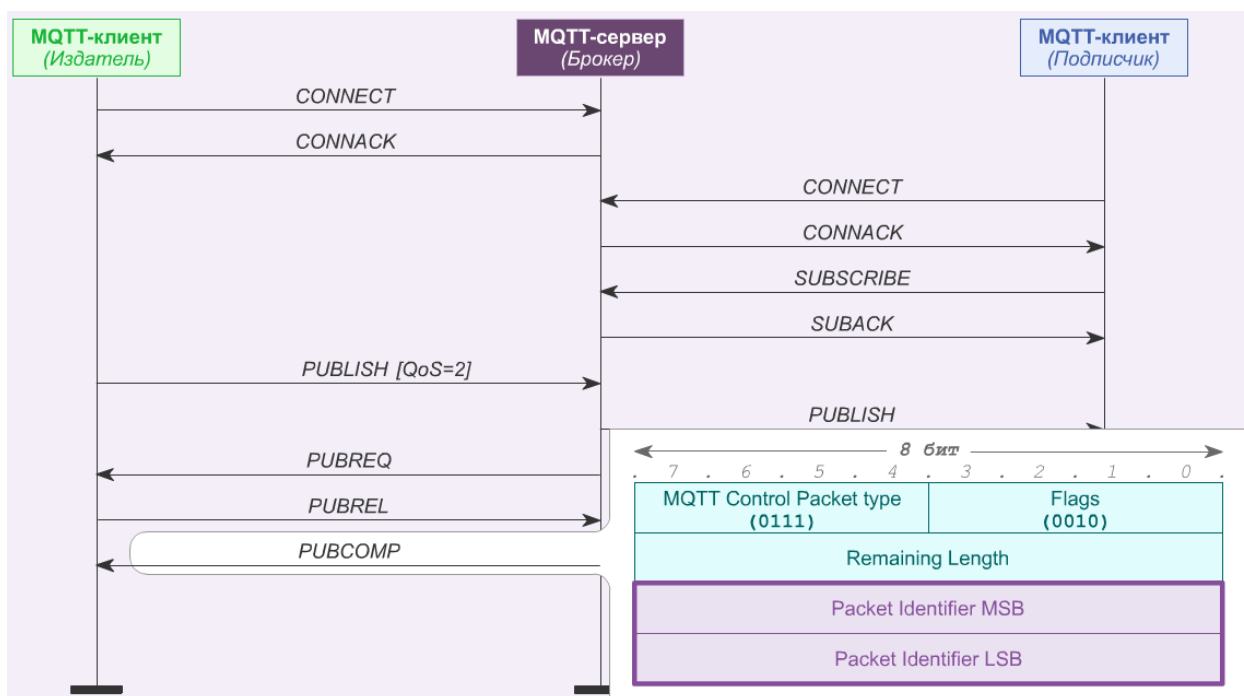
Заголовок переменной длины содержит Packet Identifier из сообщения PUBLISH, на которое отправляется подтверждение.



15. НАЗНАЧЕНИЕ СООБЩЕНИЯ PUBCOMP. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

Сообщение PUBCOMP - это ответ на сообщение PUBREL. Это четвертое и окончательное сообщение в сценарии обмена сообщениями публикаций с QoS=2. Сообщение PUBCOMP содержит заголовок фиксированной длины и заголовок переменной длины. Полезная нагрузка в сообщении PUBCOMP отсутствует.

Заголовок переменной длины содержит Packet Identifier из сообщения PUBLISH, на которое отправляется подтверждение.



16. НАЗНАЧЕНИЕ СООБЩЕНИЯ UNSUBSCRIBE. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

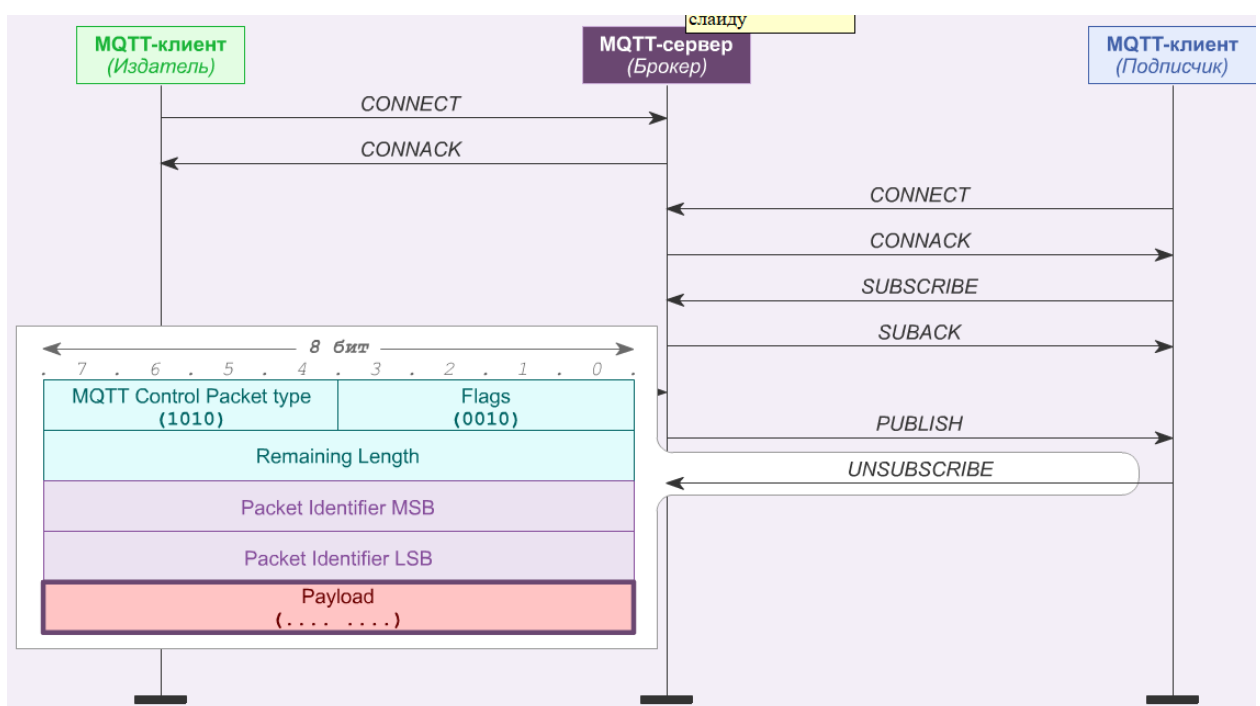
Сообщение UNSUBSCRIBE отправляется Клиентом Серверу, чтобы отписаться от тем. Сообщение UNSUBSCRIBE содержит заголовок фиксированной длины, заголовок переменной длины и полезную нагрузку.

Поле Flags

Биты 3,2,1 и 0 фиксированного заголовка сообщения UNSUBSCRIBE зарезервированы и должны быть установлены на 0,0,1 и 0 соответственно. Сервер должен рассматривать любое другое значение как неправильное и закрыть сетевое подключение.

Заголовок переменной длины содержит Packet Identifier.

Полезная нагрузка сообщения UNSUBSCRIBE содержит список фильтров тем, от которых клиент желает отписаться. Полезная нагрузка сообщения UNSUBSCRIBE должна содержать как минимум один фильтр тем.



17. НАЗНАЧЕНИЕ СООБЩЕНИЯ UNSUBACK. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

Сообщение UNSUBACK отправляется Сервером Клиенту для подтверждения получения сообщения UNSUBSCRIBE. Сообщение UNSUBACK содержит заголовок фиксированной длины и заголовок переменной длины. Полезная нагрузка в сообщение UNSUBACK отсутствует.

Заголовок переменной длины содержит Packet Identifier сообщения UNSUBSCRIBE, прием которого подтверждается

18. НАЗНАЧЕНИЕ СООБЩЕНИЯ PINGREQ. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

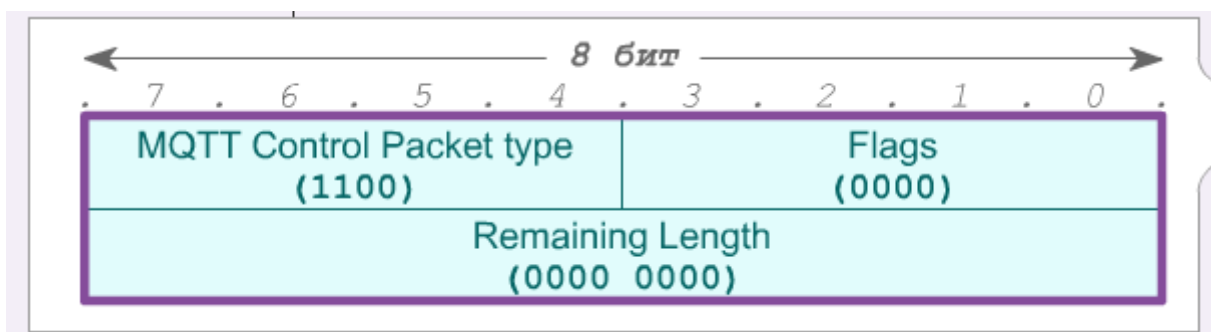
PINGREQ – PING request

Сообщение PINGREQ отправляется от клиента к серверу. Он может быть использован для того, чтобы:

- показать Серверу, что Клиент "активен" (alive) в отсутствии любых других сообщений, отправляемых от клиента к серверу.
- запросить ответ сервера, чтобы подтвердить, что он "активен" (alive).
- используя сеть, показать, что сетевое подключение активно.

Сообщение PINGREQ использует процедуру Keep Alive.

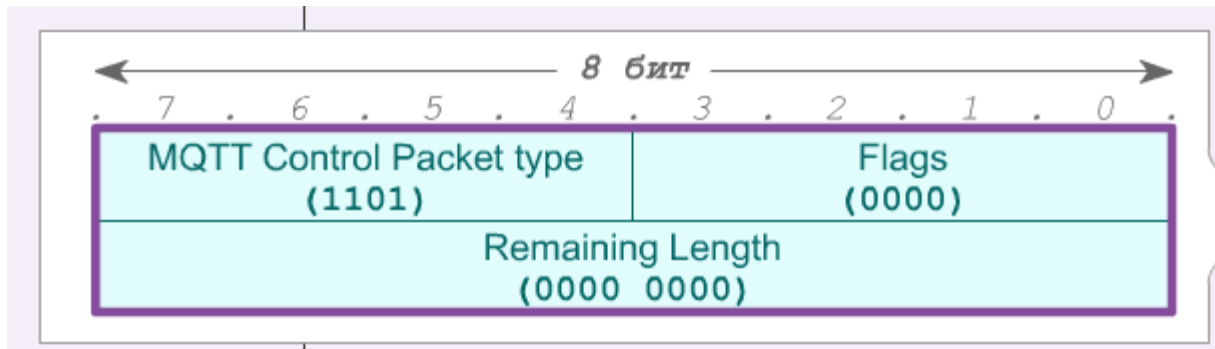
Сообщение PINGREQ содержит только заголовок фиксированной длины.



19. НАЗНАЧЕНИЕ СООБЩЕНИЯ PINGRESP. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

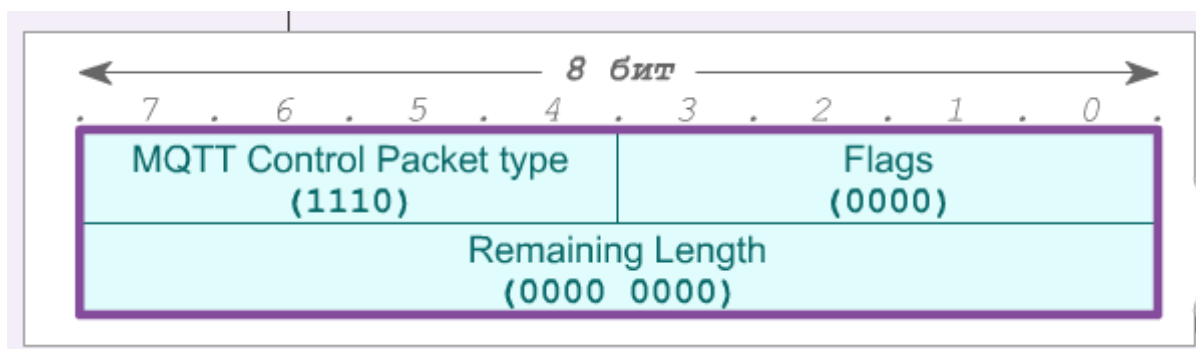
Сообщение PINGRESP отправляется сервером клиенту в ответ на сообщение PINGREQ. Это означает, что сервер "активен" (alive).

Сообщение PINGRESP содержит только заголовок фиксированной длины.



20. НАЗНАЧЕНИЕ СООБЩЕНИЯ DISCONNECT. ОПИШИТЕ ОСНОВНЫЕ ПОЛЯ СООБЩЕНИЯ И ИХ НАЗНАЧЕНИЕ.

Сообщение DISCONNECT является финальным сообщением MQTT. В представленном примере передача сообщения DISCONNECT означает, что Клиент отключился. Сообщение DISCONNECT содержит только заголовок фиксированной длины.



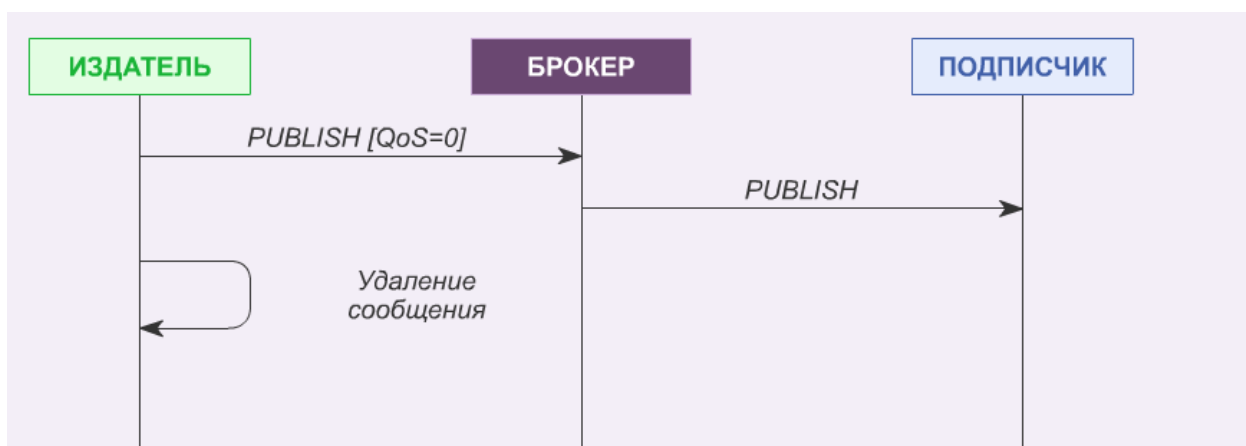
21. УРОВНИ QOS В MQTT. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ УРОВНЕЙ

Как отмечалось выше, основной отличительной чертой протокола MQTT является возможность использования различных уровней обслуживания, которые задаются значением флага QoS level. Это делает протокол MQTT более гибким, в отличие от протокола CoAP (Constrained Application Protocol), сообщения которого могут подтверждаться или обрабатываться без подтверждения.

MQTT поддерживает три уровня качества обслуживания (QoS).

- QoS 0. Этот уровень задействует стратегию «максимум однократная доставка сообщений». Приёмник сообщения не подтверждает их получение. Отправитель, соответственно, передаёт сообщение лишь раз, не предпринимая попыток по их повторной передаче.

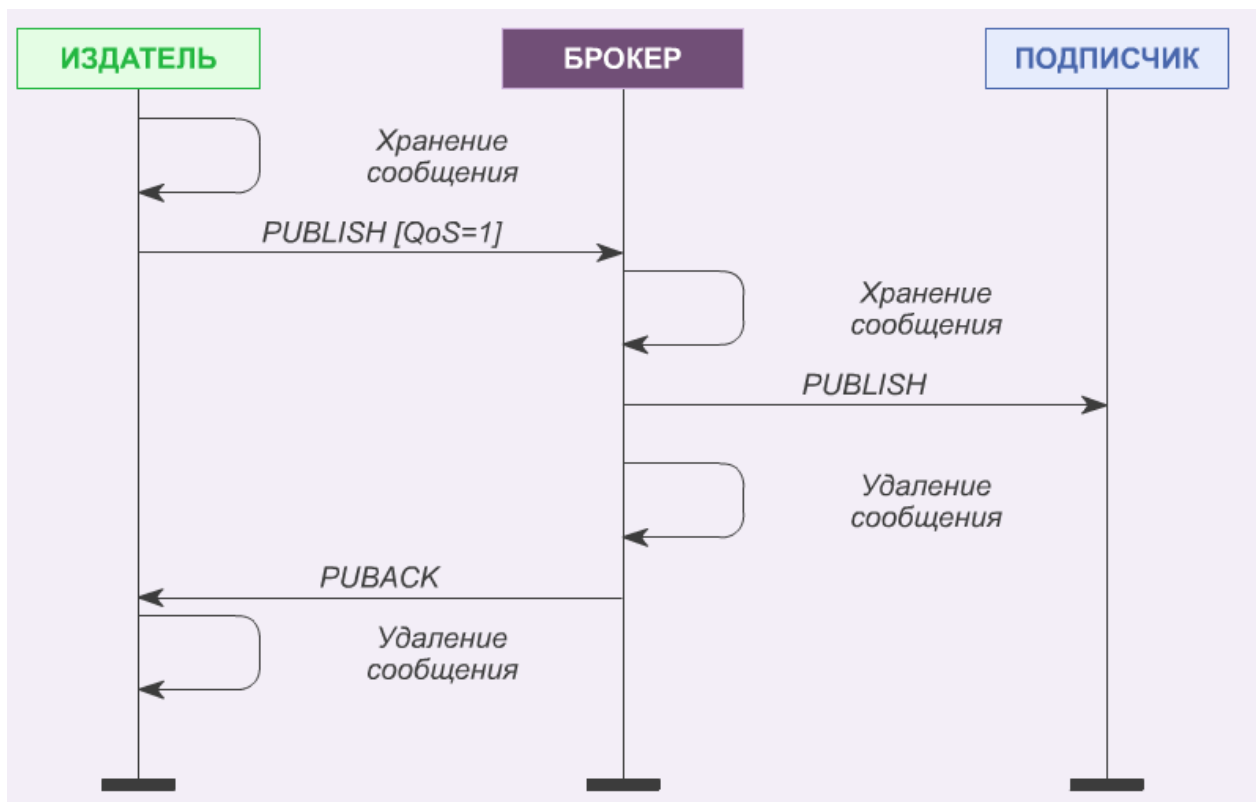
Например, Издатель публикует сообщение (PUBLISH) на брокере, а брокер публикует его для подписчика. Однако издатель не требует, чтобы это сообщение было гарантированно передано подписчику. Иными словами, подписчик может и не получить это сообщение, но это не отслеживается издателем. Описанный сценарий используется для тех случаев, когда потери данных не критичны. Например, при постоянном мониторинге температуры, когда потеря единичного измерения не играет существенной роли в общей картине.



- QoS 1. Здесь применяется подход «минимум однократная доставка сообщений». Гарантируется, что приёмник получит сообщение хотя бы один

раз. При этом подписчик может получить одно и то же сообщение несколько раз. А отправитель будет предпринимать повторные попытки отправки до тех пор, пока не получит подтверждение об успешной доставке сообщения.

Издатель публикует сообщение на брокере (PUBLISH). Брокер сохраняет это сообщение и публикует его для подписчика. Только после того, как сообщение будет опубликовано для подписчика, брокер отправляет подтверждение публикации издателю (PUBACK). То есть до тех пор, пока издатель не получит подтверждение публикации подписчику, данная публикация будет посылаться брокеру и далее подписчику. Таким образом, подписчик должен получить данное сообщение как минимум один раз.



- QoS 2. Этому уровню качества обслуживания соответствует самая медленная процедура доставки сообщений, но при этом он – самый надёжный. Его основная особенность – реализация стратегии «однократная доставка сообщений». При его использовании применяется четырёхступенчатая процедура подтверждения доставки сообщений.

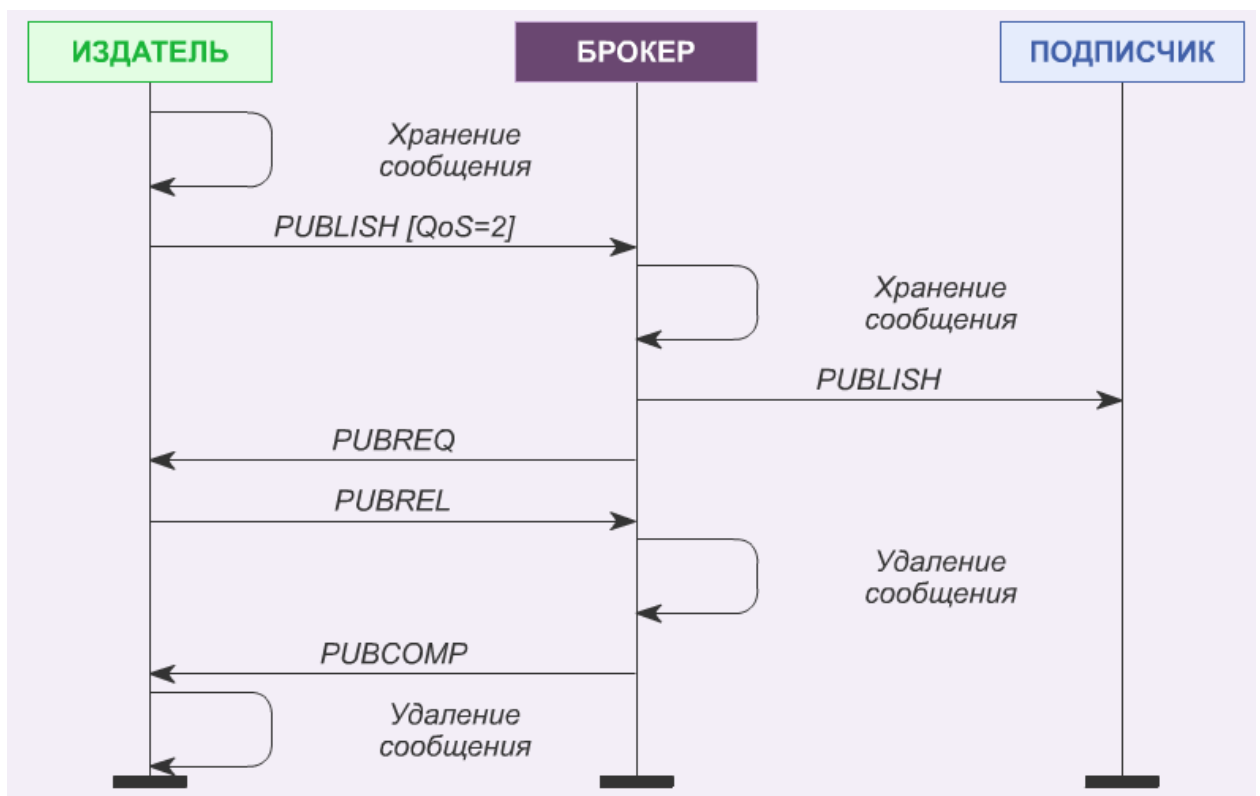
Издатель публикует сообщение на брокере (PUBLISH). Брокер сохраняет это сообщение и публикует его для подписчика. Только после того,

как сообщение будет опубликовано для подписчика, брокер отправляет подтверждение, что публикация получена (PUBREC). И Издатель инициирует процедуру завершения публикации.

счет использования дополнительных процедур подтверждения и завершения публикации (PUBREC, PUBREL, PUBCOMP). Представленный сценарий применим для ситуаций, когда нужно исключить любые потери и дублирование данных от датчиков.

Например, когда от полученного сообщения срабатывает сигнализация, вызов экстренных служб.

Выбор конкретного уровня качества обслуживания зависит от особенностей передаваемых данных и от того, насколько важно, чтобы они были доставлены.



22. СЕМАНТИКА ТЕМ В MQTT. ОДНОУРОВНЕВАЯ ТЕМА, МНОГУРОВНЕВАЯ ТЕМА. ПРИМЕР ТЕМЫ. КАК ОБЕСПЕЧИВАЕТСЯ ПОЛУЧЕНИЕ ДАННЫХ ПО НЕСКОЛЬКИМ ТЕМАМ.

Как уже говорилось ранее, клиенты (Издатели/Подписчики) подключаются к брокеру, а затем либо публикуют темы и сообщения в них, либо подписываются на темы и получают сообщения, в этих темах опубликованные.

Темы представляют собой символы с кодировкой UTF-8. В протоколе MQTT принята иерархическая структура построения тем (формат «дерева»), что значительно упрощает их организацию и доступ к данным.

Темы состоят из одного или нескольких уровней, которые разделены между собой символом «/».

Подписчик может так же получать данные сразу по нескольким темам, для этого существуют wildcard символы. Они бывают двух типов: одноуровневые и многоуровневые.



Для более простого понимания рассмотрим в примерах каждый из них:

- Одноуровневый wildcard символ. Для его использования применяется символ «+» .

К примеру, необходимо узнать в каких жилых комнатах горит свет. Для этого выполняем запрос `/home/living-space/+/light` и получаем ответ с двух датчиков света установленных в жилых комнатах.

Запрос:

`/home/living-space/+/light`

Ответ:

`/home/living-space/room_1/light`

`/home/living-space/room_2/light`

- Многоуровневый wildcard символ. Для его использования применяется символ «#»

К примеру, чтобы получить данные с различных датчиков всех спален в доме мы выполняем запрос `/home/living-space/#` и в ответ получаем данные с 2-х датчиков одной комнаты и 3-х датчиков другой комнаты.

Запрос:

`/home/living-space/#`

Ответ:

`/home/living-space/room_1/light`

`/home/living-space/room_1/temp`

`/home/living-space/room_2/light`

`/home/living-space/room_2/temp`

`/home/living-space/room_2/humidity`

23. ПРОБЛЕМЫ MQTT

Протокол TCP разрабатывался для устройств, имеющих больше памяти и вычислительной мощности, чем большинство устройств IoT, имеющих небольшие размеры и с ограниченную мощность. Протокол TCP требует предварительного установления соединения (процедура «рукопожатия») для передачи сообщений. Это увеличивает время включения и установления связи, что уменьшает время работы батареи питания. TCP-подключенные устройства, как правило, держать сокет открытыми друг для друга с постоянным сеансом. Это добавляет к требованиям к мощности и объему памяти устройства.

Брокер может повлиять на масштабируемость сети, поскольку каждое подключенное устройство создает дополнительную нагрузку. Сеть может расти настолько, насколько может поддерживать Брокер.

Например, в случае сбоя питания устройства Iot (издатели и подписчики) будут продолжать работать за счет батарей питания, а Брокер будет отключен. Функционирование всей сети будет невозможно до тех пор, пока не возобновится питание Брокера.

Сообщения MQTT по умолчанию не зашифрованы. Это делает его изначально незащищенным и требует дополнительные меры для организации безопасности.