С развитием промышленности увеличивается количество устройств, которые нужно контролировать и получать от них различные данные. Для решения проблем взаимодействия большого количества устройств и проблем объединения устройств в одну сеть была создана концепция Интернета вещей (англ. Internet of Things, IoT) – это когда устройства объединяются по какому-то признаку в одну сеть, потом уже несколько подобных сетей объединяются в другую большую сеть и так далее.
Устройства в таких сетях взаимодействуют друг с другом по средствам различных интерфейсов и протоколов передачи данных. Так как мы говорим о промышленном применении концепции IoT, в которой должны использоваться промышленное оборудование со своими протоколами и аппаратными средствами, то мы переходим к концепции IIoT (Промышленного Интернета вещей).
Промышленный интернет вещей (IIoT) – новый шаг в развитии промышленной автоматизации. Узнайте больше о решениях, предлагаемых компанией IPC2U в сфере IIoT
Для взаимодействия между собой устройства используют различные промышленные протоколы, одним из популярных протоколов для этой цели является MQTT.

Что такое MQTT?

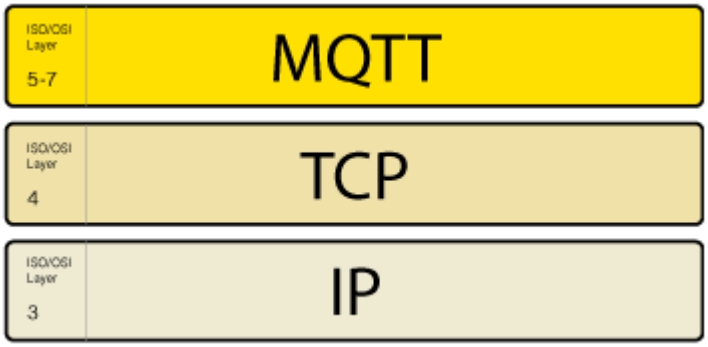
MQTT или Message Queue Telemetry Transport – это легкий, компактный и открытый протокол обмена данными созданный для передачи данных на удалённых локациях, где требуется небольшой размер кода и есть ограничения по пропускной способности канала.
Вышеперечисленные достоинства позволяют применять его в системах M2M (Машинно-Машинное взаимодействие) и IIoT (Промышленный Интернет вещей).
Также существует версия протокола MQTT-SN (MQTT for Sensor Networks), ранее известная как MQTT-S, которая предназначена для встраиваемых беспроводных устройств без поддержки TCP/IP сетей, например, Zigbee.

Особенности протокола MQTT

Основные особенности протокола MQTT:

- Асинхронный протокол
- Компактные сообщения
- Работа в условиях нестабильной связи на линии передачи данных
- Поддержка нескольких уровней качества обслуживания (QoS)
- Легкая интеграция новых устройств

Протокол MQTT работает на прикладном уровне поверх TCP/IP и использует по умолчанию 1883 порт (8883 при подключении через SSL).



Обмен сообщениями в протоколе MQTT осуществляется между клиентом (client), который может быть издателем или подписчиком (publisher/subscriber) сообщений, и брокером (broker) сообщений (например, Mosquitto MQTT).
Издатель отправляет данные на MQTT брокер, указывая в сообщении определенную тему, топик (topic). Подписчики могут получать разные данные от множества издателей в зависимости от подписки на соответствующие топики.
Устройства MQTT используют определенные типы сообщений для взаимодействия с брокером, ниже представлены основные:

- Connect – установить соединение с брокером
- Disconnect – разорвать соединение с брокером

- Publish – опубликовать данные в топик на брокере
- Subscribe – подписаться на топик на брокере
- Unsubscribe – отписаться от топика

Схема простого взаимодействия между подписчиком, издателем и брокером

Наверх к оглавлению

Семантика топиков

Топики представляют собой символы с кодировкой UTF-8. Иерархическая структура топиков имеет формат «дерева», что упрощает их организацию и доступ к данным. Топики состоят из одного или нескольких уровней, которые разделены между собой символом «/».
Пример топика в который датчик температуры, расположенный в спальной комнате публикует данные брокеру:
/home/living-space/living-room1/temperature
Подписчик может так же получать данные сразу с нескольких топиков, для этого существуют wildcard. Они бывают двух типов: одноуровневые и многоуровневые. Для более простого понимания рассмотрим в примерах каждый из них:

- Одноуровневый wildcard. Для его использования применяется символ «+»
  К примеру, нам необходимо получить данные о температуры во всех спальных комнатах:
  /home/living-space/+/temperature
  В результате получаем данные с топиков:
  /home/living-space/living-room1/temperature
  /home/living-space/living-room2/temperature
  /home/living-space/living-room3/temperature
- Многоуровневый wildcard. Для его использования применяется символ «#»
  К примеру, чтобы получить данные с различных датчиков всех спален в доме:
  /home/living-space/#
  В результате получаем данные с топиков:
  /home/living-space/living-room1/temperature
  /home/living-space/living-room1/light1
  /home/living-space/living-room1/light2
  /home/living-space/living-room1/humidity
  /home/living-space/living-room2/temperature
  /home/living-space/living-room2/light1
  …

Структура сообщений

MQTT сообщение состоит из нескольких частей:

- Фиксированный заголовок (присутствует по всех сообщениях)
- Переменный заголовок (присутствует только в определенных сообщениях)
- Данные, «нагрузка» (присутствует только в определенных сообщениях)

Фиксированный заголовок

Message Type – это тип сообщения, например: CONNECT, SUBSCRIBE, PUBLISH и другие.
Flags specific to each MQTT packet – эти 4 бита отведены под вспомогательные флаги, наличие и состояние которых зависит от типа сообщения.
Remaining Length – представляет длину текущего сообщения(переменный заголовок + данные), может занимать от 1 до 4 байта.
Всего в протоколе MQTT существует 15 типов сообщений:

Тип сообщения	Значение	Направление передачи	Описание
Reserved	0000 (0)	нет	Зарезервирован
CONNECT	0001 (1)	К* -> С**	Запрос клиента на подключение к серверу
CONNACK	0010 (2)	К <- С	Подтверждение успешного подключения
PUBLISH	0011 (3)	К <- С, К -> С	Публикация сообщения
PUBACK	0100 (4)	К <- С, К -> С	Подтверждение публикации
PUBREC	0101 (5)	К <- С, К -> С	Публикация получена
PUBREL	0110 (6)	К <- С, К -> С	Разрешение на удаление сообщения
PUBCOMP	0111 (7)	К <- С, К -> С	Публикация завершена
SUBSCRIBE	1000 (8)	К -> С	Запрос на подписку
SUBACK	1001 (9)	К <- С	Запрос на подписку принят
UNSUBSCRIBE	1010 (10)	К -> С	Запрос на отписку
UNSUBACK	1011 (11)	К <- С	Запрос на отписку принят
PINGREQ	1100 (12)	К -> С	PING запрос
PINGRESP	1101 (13)	К <- С	PING ответ
DISCONNECT	1110 (14)	К -> С	Сообщение об отключении от сервера
Reserved	1111 (15)		Зарезервирован

*К – клиент, **С – сервер
Флаги
Четыре старших бита первого байта фиксированного заголовка отведены под специальные флаги:

Bit	7	6	5	4	3	2	1	0
Byte 1	Message type				DUP	QoS	QoS	Retain
Byte 2	Remaining Length							

DUP – флаг дубликата устанавливается, когда клиент или MQTT брокер совершает повторную отправку пакета (используется в типах PUBLISH, SUBSCRIBE, UNSUBSCRIBE, PUBREL). При установленном флаге переменный заголовок должен содержать Message ID (идентификатор сообщения)
QoS – качество обслуживания (0,1,2)
RETAIN – при публикации данных с установленным флагом retain, брокер сохранит его. При следующей подписке на этот топик брокер незамедлительно отправит сообщение с этим флагом. Используется только в сообщениях с типом PUBLISH.

Переменный заголовок

Переменный заголовок содержится в некоторых заголовках.
В нём помещаются следующие данные:

- Packet identifier – идентификатор пакета, присутствует во всех типах сообщений, кроме: CONNECT, CONNACK, PUBLISH(с QoS <1), PINGREQ, PINGRESP, DISCONNECT
- Protocol name – название протокола (только в сообщениях типа CONNECT)
- Protocol version – версия протокола (только в сообщениях типа CONNECT)
- Connect flags – флаги указывающие на поведение клиента при подключении

Bit	7	6	5	4	3	2	1	0
Byte 8	User name	Password	Will Retain	Will QoS		Will Flag	Clean Session	Reserved

User name – при наличии этого флага в «нагрузке» должно быть указано имя пользователя (используется для аутентификации клиента)
Password – при наличии этого флага в «нагрузке» должен быть указан пароль (используется для аутентификации клиента)
Will Retain – при установке в 1, брокер хранит у себя Will Message.
Will QoS– качество обслуживания для Will Message, при установленном флаге Will Flag, Will QoS и Will retain являются обязательными.
Will Flag - при установленном флаге, после того, как клиент отключится от брокера без отправки команды DISCONNECT(в случаях непредсказуемого обрыва связи и т.д.), брокер оповестит об этом всех подключенных к нему клиентов через так называемый Will Message.
Clean Session – очистить сессию. При установленном «0» брокер сохранит сессию, все подписки клиента, а так же передаст ему все сообщения с QoS1 и QoS2, которые были получены брокером во время отключения клиента, при его следующем подключении. Соответственно при установленной «1», при повторном подключении клиенту будет необходимо заново подписываться на топики.

- Session Present – применяется в сообщении с типом CONNACK. Если брокер принимает подключение с Clean Session = 1 он должен установить «0» в бит Session Present(SP). Если брокер принимает подключение с Clean Session = 0, то значение бита SP зависит от того, сохранял ли брокер ранее сессию с этим клиентом (если так, то в SP выставляется 1 и наоборот). То есть этот параметр позволяет клиенту определить была ли сохранена брокером предыдущая сессия.
- Connect Return code – если брокер по каким то причинам не может принять правильно сформированный CONNECT пакет от клиента, то во втором байте CONNACK пакета он должен установить соответствующее значение из нижеуказанного списка:

Значение	Возвращенное значение	Описание
0	0x00 Connection Accepted	Подключение принято
1	0x01 Connection Refused, unacceptable protocol version	Брокер не поддерживает версию протокола, используемую клиентом
2	0x02 Connection Refused, identifier rejected	Client ID подключаемого клиента нет в списке разрешенных
3	0x03 Connection Refused, Server unavailable	Соединение установлено, но MQTT сервис не доступен
4	0x04 Connection Refused, bad user name or password	Не правильный логин или пароль
5	0x05 Connection Refused, not authorized	Доступ к подключению запрещен
6-255		зарезервировано

- Topic Name – название топика

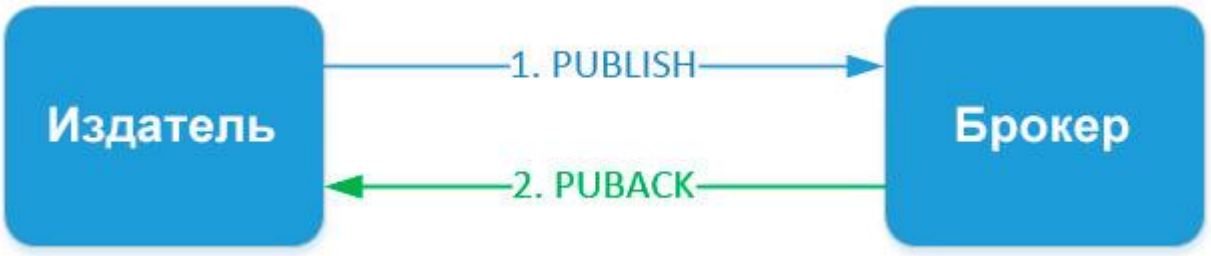
Данные, нагрузка

Содержание и формат данных, передаваемых в MQTT сообщениях, определяются в приложении. Размер данных может быть вычислен путём вычитания из Remaining Length длины переменного заголовка.

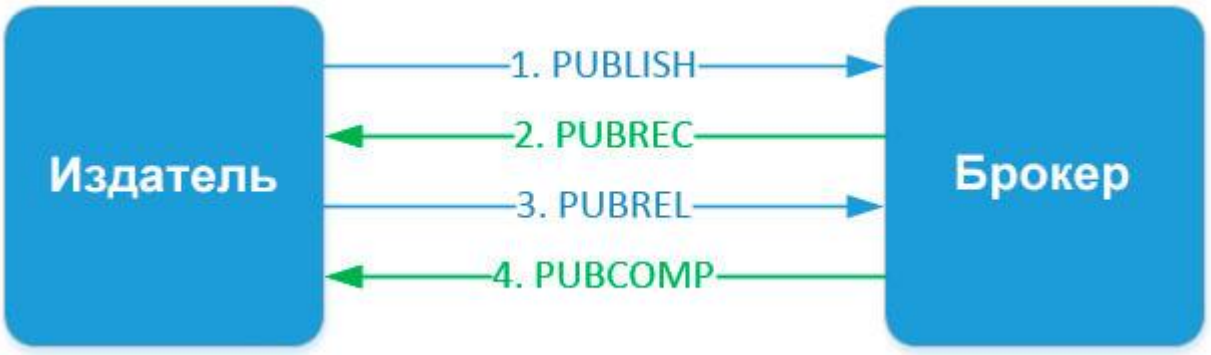
Качество обслуживания в протоколе MQTT (QoS)

MQTT поддерживает три уровня качества обслуживания (QoS) при передаче сообщений.
QoS 0 At most once. На этом уровне издатель один раз отправляет сообщение брокеру и не ждет подтверждения от него, то есть отправил и забыл.

QoS 1 At least once. Этот уровень гарантирует, что сообщение точно будет доставлено брокеру, но есть вероятность дублирования сообщений от издателя. После получения дубликата сообщения, брокер снова рассылает это сообщение подписчикам, а издателю снова отправляет подтверждение о получении сообщения. Если издатель не получил PUBACK сообщения от брокера, он повторно отправляет этот пакет, при этом в DUP устанавливается «1».



QoS 2 Exactly once. На этом уровне гарантируется доставка сообщений подписчику и исключается возможное дублирование отправленных сообщений.



Издатель отправляет сообщение брокеру. В этом сообщении указывается уникальный Packet ID, QoS=2 и DUP=0. Издатель хранит сообщение неподтвержденным пока не получит от брокера ответ PUBREC. Брокер отвечает сообщением PUBREC в котором содержится тот же Packet ID. После его получения издатель отправляет PUBREL с тем же Packet ID. До того, как брокер получит PUBREL он должен хранить копию сообщения у себя. После получения PUBREL он удаляет копию сообщения и отправляет издателю сообщение PUBCOMP о том, что транзакция завершена.

Защита передачи данных

Для обеспечения безопасности в MQTT протоколе реализованы следующие методы защиты:

- Аутентификация клиентов. Пакет CONNECT может содержать в себе поля USERNAME и PASSWORD. При реализации брокера можно использовать эти поля для аутентификации клиента
- Контроль доступа клиентов через Client ID
- Подключение к брокеру через TLS/SSL