MQTT

This Decument is written By Mahmoud Badr

IoT

We can say that IoT is connecting anything anywhere any time.
IoT is used to Monitoring and controlling
CIoT: cognitive Internet of Things =IoT+AI same as IoT but it can make decisions by it self

IoT protocols:

Protocol	Meaning	Layer	Transport protocol
COAP	Constraine Access Protocol	App	UDP
MQTT	message Queuing Telemetry Transport	App	TCP

Transport Layer Protocol:

1. TCP: connection-oriented protocol
2. UDP: connectionless protocol
   -> in Embedded system we use MQTT as it's a light weight messaging protocol

MQTT

MQTT work in publish-Subscribe -- pub-sub -- Archeticture in client server model

- In MQTT we like to call the server as broker
- We have some definitions to know:

1. Publish: to give the data to broker in a specific topic
2. Subscibe: to get the data from specific topic in the broker
3. Topic: the container of data
4. Broker: the broker between subscriber and puplisher

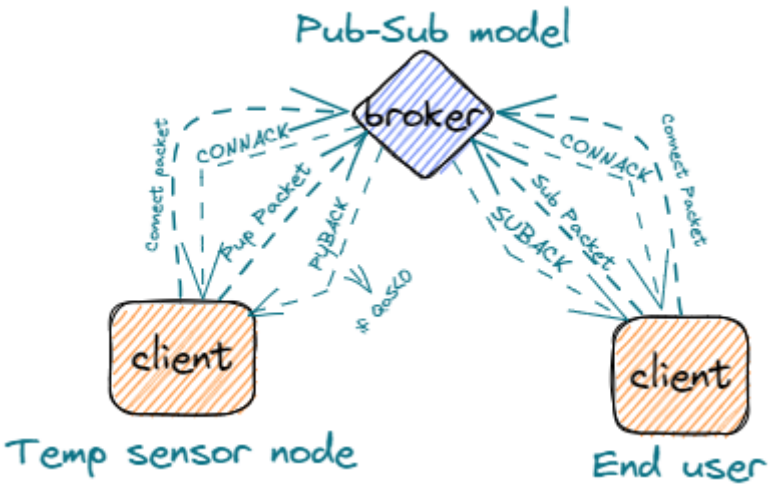
Send Data

we will asume that we have temp sensor node -- mcu connected to the internet -- and end user need to know the temp sensor reading

- First connect to the broker with TCP and IP and port number -- Impelemented in the ESP --
- We connect to the broker by Ethernet, GSM or wifi and know we can start send the Packets
- For security the broker must identifiy the clients connected to it so, the first packet to send is the connect packet.
- The connect packet must be sent immediately after opening the TCP connection.
- The connect Packet is sent by the node and end user and CONNACK will be the reply
- The connect packet has some connection parameters from which the id of node or user.
- now what to do first publish or subscribe?
- If we publish first some information will be discarded so we must subscribe first
- so we send the subscribe packet so he can subscribe for the topic he want -- ex: temp topic -- and broker will reply with SUBACK.
- Then the puplisher will puplish to the topic he want -- ex: temp topic -- and broker will reply or not depending on Quality of service -- QoS --
- we have three QoS:
  1. QoS 0: Fire and forget -- At most once --:
     - No Ack is sent from the broker
     - send the packet and forget it
     - If the message is sent it will be sent once at most
  2. QoS 1: At least once:
     - The broker send PUBACK
     - The packet may sent more than once due to trafic in network
     - The packets here is numbered with packet Identifier so that if the packet is sent more than one time the MQTT set a bit called duplicate to tell the broker that its not the first to time to send this so, the broker discard it.
  3. QoS 2: Exactly once:
     - The broker send more than one ACK
       - PUBACK: when the broker received the message
       - PUBREC: ensure that publish is done
       - PUBREL: the Packet is sent to subscribers.

> - In embedded system we use QoS 0 if the information isn't important and use Qos 1 if the information is important

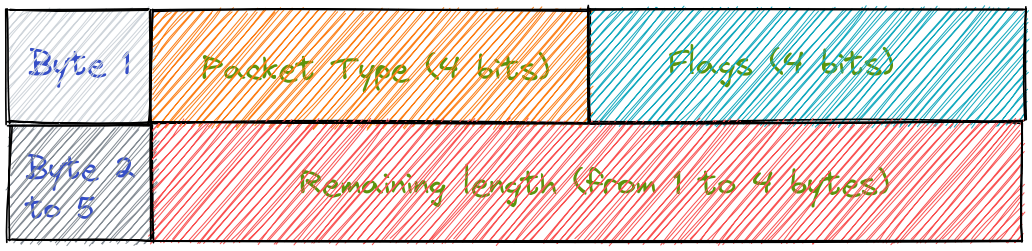
- The subscriber must tell the broker max supported QoS



MQTT Packets

- All the packets we want to send is called control packets
- MQTT control packets consist of:

1. Fixed Header: Presented in all Packets
   - Consist of 2 to 5 bytes
   - The first byte divided in to two equal sections each of four bits
     - b0 to b3 are flags specific to each control packet
     - b4 to b7 are the packet type to be sent
   - The remianing bytes is used to identify the remaining length of packet and it's variable length bytes as it can be from 1 to 4 bytes. How?
     - Rem Length = len of var header + payload
     - Total length = Rem len + len of Rem len + 1
     - The MSB is reserved as a flag to identify that the byte is for the remainging length
       - if the MSB is set in the byte then the next byte is with us
       - So the byte can carry up to 127 byte as we have just 7 bits and the MSB as a flag



2. Variable Header: in some packets
   - Most popular used in variable header is packet identifier

- Packet id used to have a flow control Mechanism works as a sequence number for packets to identify the arrival of packet from client to broker and vice versa
- The packet Identifier consist of two bytes starting from 1.

- Used when QoS > 0 for Publish packets.

3. Payload: in some packets

- So, what are the packets we need to know

1. Connect Packet:
   - After the TCP connection is established, The first packet to be sent is the connect packet
   - If the client send a second connect packet before receiving Ack the broker disconnect him
   - For the Packet:
   - Fixed Header

Byte number	value	comment
1	0x10	The type is 1 and the flags are reserved
2		The Rem length

> - Rem length = 10 + 2 + strlen(ClientId) -- For puplic broker --

   - Variable Header

Byte number	value	comment
1 : 2	0x0004	The protocol name size
3 : 6	MQTT	The protocol name
7	0x04	protocol level byte
8	0x02	Connect Flags
9 : 10	0xFFFF	Keep Alive

- Connect flags:

Bit number	Usage	Comment

Bit number	Usage	Comment
7	Username flag	set if using private broker
6	password flag	set if using private broker
5	Will Retain	save the will message
4 : 3	Qos Retain	send the will message with Qos x
2	Will flag	set the flag when there is a will message
1	clean session	Clean the last messages sent when I was offline
0	Reserved	Must be 0

Notes:

- if you use private broker you must have the preconfigured username and password by broker admistrator and set the username and password flags.
- The will message is sent by the broker if there is an error with broker
- The keep Alive time is the time to be connected to the broker in case of not sending any messages and consist of two bytes for max 18 hours, 12 minutes, and 15 seconds.
- if you want to keep connection you can send ping Request and ping response packets
  - PING REQUEST: From client broker to tell him I am online
    - consist of the Fixed Header 2 bytes: 0xC000
  - PING RESPONSE: From broker to client that your KAT is renewed:
    - consist of the Fixed Header 2 bytes: 0xD000
- The strings is sent as UTF-8 Encoded and that means that:
  - This string is ASCII represented
  - Any string will be sent will has two bytes before it identify the strlen(MSB , LSB).
- any two bytes describe length of any thing the LSB is the second Byte

- Payload:

Byte number	value	Comment
1 : 2	0x00strlen(ID)	The first two bits for the length of the client id
3 : n	ID	the client id string
n+1 : n+2	Username len	The user name length

Byte number	value	Comment
n+3 ...	username	the username as string and the password after it as username

> - Response: The response is the connect Ack

---

1. Connect Ack -- CONNACK -- packet:
   - Fixed Header

Byte number	value	comment
1	0x20	The type is 1 and the flags are reserved
2	0x02	The Rem length

   - Variable Header:

Byte number	value	comment
1	0x00	Reserved but the LSB is for the peresent sessions
2	0x00	Connect return code

> - The connect return code has some states

return	state
0x00	Connection accepted
0x01	Server doesn't support level of the MQTT protocol requested by the Client
0x02	The Client identifier is correct UTF-8 but not allowed by the Server
0x03	the MQTT service is unavailable

---

1. publish packet
   - Fixed Header:

Byte number	Value	Comment
1	0x31 -- 0x33	The frame type and use the QoS as 0 - 1
2		Remaining Length

- Rem Len = 2 + strlen(Topic name) + len(Message) + (2 if QoS >0)
- The flags:

falg	Set	Reset
DUP	re-delivery of the pub packet	The first trail to send the packet
QoS	discussed before	-
RETAIN	The server save the packet until the subscriber need it	The server saves the last recently sent packet before it

- Variable Header:
  - Topic name

Byte number	comment
1 : 2	The topic name length
2 : n	The topic name

  - Packet identifier: used when QoS >0 used as 2 bytes for it.
- Payload:
  - The payload is the message to be published
- Response: there is a response when QoS > 0

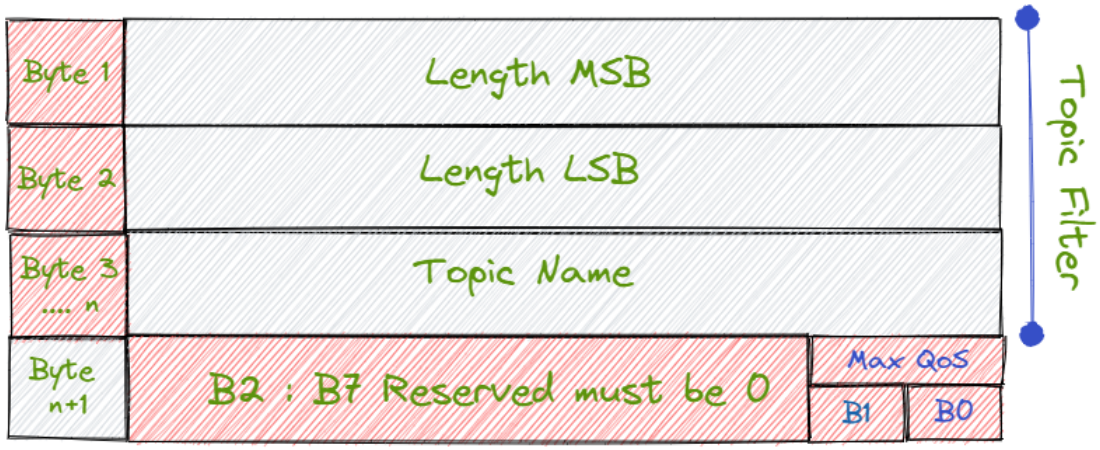
2. Publish Acknowledge:
   - Used when Qos > 0 and sent when the pub is correct and acknowledged
   - Fixed Header

Byte number	value	comment
1	0x40	The type is 4 and the flags are reserved
2	0x02	The Rem length

   - Variable Header:
     - Packet identifier: 2 bytes contains the packet identifier
3. Subscribe Packet:
   - Fixed Header:

Byte number	Value	Comment
1	0x82	The frame type and flags are reserved
2		Remaining Length

    - Rem Len = 2+2+len(Topic)+1
- Variable Header:
    - Packet identifier: 2 bytes contains the packet identifier
    - The sub packet has Ack so the Packet id is a must
        - Note: why the connect packet doesn't has id like sub?
        - As the sub packet may be sent at runtime any time but the connect packet is sent only at first time as the first packet.
- Payload
    - consist of topic filters -- a UTF-8 encoded strings --
    - each topic filter is followed by encoded byte contains the max supported QoS.



- What if:
    - A Server receives a SUBSCRIBE Packet containing a Topic Filter that is identical to an existing Subscription Topic Filter?
        - It MUST completely replace that existing Subscription with a new 968 Subscription. The Topic Filter in the new Subscription will be identical to that in the previous Subscription, although its maximum QoS value could be different. Any existing retained messages matching the Topic Filter MUST be re-sent, but the flow of publications MUST NOT be interrupted.
    - A Server receives a SUBSCRIBE packet that contains multiple Topic Filters?
        - It MUST handle that packet as if it had received a sequence of multiple SUBSCRIBE packets, except that it combines their responses into a single SUBACK response.
- Response: a suback is sent by broker and the Server is permitted to start sending PUBLISH packets matching the Subscription before the Server 964 sends the SUBACK Packet.

4. Subscribe Acknowledge:
    - Fixed Header

Byte number	value	comment
1	0x90	The type is 4 and the flags are reserved
2		The Rem length

    - Rem Len = 2 + Num of topic filters being Acked
- Variable Header:
    - Packet identifier: 2 bytes contains the packet identifier
- Payload:
    - Contains the Return codes of topic filters subscribed and came in the order of Topic filter in Subscribe Packet.
Return Code	description
0x00	Success - Maximum QoS 0
0x01	Success - Maximum QoS 1
0x02	Success - Maximum QoS 2
0x80	Failure

MQTT Functions

```
typedef enum
{
    MQTT_QOS0=0,
    MQTT_QOS1,
    MQTT_QOS2
}Mqtt_Qos_t;

typedef struct
{
    u8* TopicName;
    u8* Msg;
    Mqtt_Qos_t Qos;
    u32 MsgLen;
}Mqtt_Publish_t;

ErrorState_t Mqtt_Connect(u8* Copy_ClientId);

ErrorState_t Mqtt_Publish(Mqtt_Publish_t* Copy_PubPacket);

ErrorState_t Mqtt_Subscribe(u8* TopicName, u8* Data);
```

Advices

1. Learn java, oop, python and make tools using them
2. You mustn't test every thing everytime, why not making a tool and let it make the test
3. To be better than you instructor read and impelement more than him

Thanks .. Mahmoud Badr