**Message Queuing Telemetry Transport (MQTT)**

Arthur Bondar  
NSCC Applied Energy Research  
December 15, 2017

Contents

[TCP 2](#_Toc501107576)

[Broker 2](#_Toc501107577)

[Topic 3](#_Toc501107578)

[Publishing 3](#_Toc501107579)

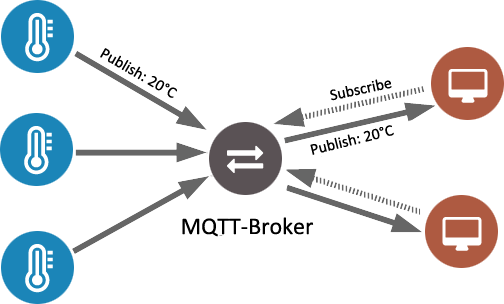
[Subscribing 3](#_Toc501107580)

[DNS 4](#_Toc501107581)

[Summary 4](#_Toc501107582)

[Using MQTT mobile apps 5](#_Toc501107583)

Message Queuing Telemetry Transport (MQTT) was developed by IBM. It is a lightweight internet protocol used for machine to machine communication, and it is central to the Internet of Things. MQTT messaging protocol is also used by Facebook for its chat services.



TCP   
MQTT protocol is built upon the Transport Layer of the OSI model – TCP. TCP/IP is the most fundamental internet protocol for data exchange. TCP creates a “Session” between two computers where information is exchanged. When one device requests a session and sends the first packet of information, the second device determines its IP address and is able to respond with more packets. If the Session is not maintained (kept “alive”) during a period of a few minutes, the link is between devices disappears. When one device initiates a session from a closed network, it is registered by the main switch as outgoing traffic, and thus it is not blocked. When a device responds back to a closed network, traffic is not registered as unknown incoming traffic because the session was initiated from within the closed network.

Broker Broker is a central computer on MQTT network, to which all the information is being sent. Although MQTT enables communication between devices, the link is not direct. In fact, each device is talking ONLY to the broker, and it is broker’s responsibility to forward messages. An analogy can be drawn between a magazine/newsletter distributer. Like the magazine delivery service, broker is only responsible for distributing new information to its clients, and keep track of which client is subscribed to which topic.   
Broker can be any single-board (raspberry pi) or any personal computer that has a broker program installed and have internet access. Broker does not keep records of IP addresses for the every device in the IoT network. Devices talk to the broker by initiating a TCP Session. Once the session is created, broker can respond to that IP address and receive incoming data, or transmits any data that needs to be forwarded. The fact that devices are the ones who initiate the TCP Session, allows MQTT to be very versatile. Since the traffic from the device to the broker is registered as outgoing traffic, it’s allowed in most networks, including mobile.   
Another major advantage of MQTT structure is that devices don’t need to know each other’s IP addresses. IP addresses for non-commercial internet users change constantly, and it would not be possible to keep track any specific IP address. MQTT solves this problem by needing to know only one address – broker’s address.   
For our application, broker software is installed on the same computer as our Linux server, and the address is *aerlab.ddns.net*. We are using open source broker program is called *Mosquito*.  
Note: MQTT operates on port 1883, this port must be enabled in the main network switch.

TopicEach message in MQTT consists of two parts: Topic and Payload (payload is the message data). Similarly to an email topic, topics in MQTT are used to briefly describe the message. Topics are broken down into categories and sub-categories, in a similar fashion to a file directory. The topic is sent with the message in form of a single string separated by ‘/’ line.   
Here is an example of a topic for aquarium water temperature:   
*Home/LivingRoom/Aquarium/Temperature*Using the magazine analogy, topic is the name of the magazine that is being distributed by the broker. Developers and Users can choose any topics for their messages.

# Publishing

Any devices on MQTT network can publish or subscribe to any specific topic. Publishing data to MQTT network is as simple as choosing a topic and writing a message. When a device publishes data to the networks, it does not specify any address for data to be sent. Topic is the only address for that specific information. This information is being broadcasted across all MQTT network, in a similar fashion to a group chat conversation. Usually, publishers are devices that collect data. They have preselected topics, to which, they continuously post information.   
In our application sensor modules are publishers. We are using *PubSubClient* library to publish sensor data on ESP microcontrollers. Additionally, free mobile apps are available for both platforms that allow publishing and subscribing to topics.

# Subscribing

When device is subscribed to a particular topic, all messages with that topic will be forwarded to it by the broker. When a device connects to the broker, and subscribes to a topic, a TCP Session is opened between that device and the broker. Behind the scenes, this session is maintained “alive” by sending confirmation requests between the device and the broker. While the session is open, if a message comes with the same topic as the one device is subscribed to, it is being forwarded by the broker. As far as the broker concerned, it only needs to forward messages between publishers and subscribers. A single device can publish or subscribe to multiple topics.   
This messaging structure allows adding modules very easily. Continuing with the example from the topic section, if we have an aquarium temperature monitor in the living room that publishes data. To be able to view the data from your phone you simply need to download MQTT phone app and subscribe to that topic. And if you want to store the data as well, subscribe to the same topic from a computer that has a database installed.

In our application, our Linux server is subscribed to all data topics and is able to publish certain data as well (sensor sample rate). We are using *EclipsePaho* library to subscribe and log data into the database, and Node-RED MQTT node to display data in the user interface.

# DNS

Domain Name System (DNS) associates IP addresses with human readable text strings. In short, it is responsible for translating a human-readable URL like *www.google.com* into an actual internet address *172.217.9.238*. Every user who browses on the internet have an IP address, however, since they do not provide any internet services, their IP is temporary. It is assigned by the Internet Service Provider and is often subject to change. To provide internet services a few things are needed. First, receive a static IP address from internet service provider. Second is to purchase a Domain Name, which means, register your static IP address on the DNS network and choose a name.  
In our application, we are using Dynamic DNS system. We are hosting our server in a remote location and do not have static IP address nor domain name. Dynamic DNS works around the problem of not having static IP address by constantly keeping track of the current IP address. When Dynamic DNS client is installed on a computer or a router, every time that ISP gives a new IP to that device, this IP is being sent to the Dynamic DNS provider. The provider links that IP with a domain name, and thus, you can reach a web server even though it’s IP has been changed.   
For example, when you request *aerlab.ddns.net* web page, the computer first connects to DDNS provider’s database, where it receives the last IP address that is associated with that domain name.   
We are using [*noip.com*](https://www.noip.com/) Dynamic DNS provider.  
Note: DNS is not a requirement for MQTT communication. MQTT can happen within the local network, provided that port 1883 is enabled. Having Domain Name simply expands the reach of the network and allows connection worldwide.

# Summary

MQTT is a simple messaging protocol in which devices establish TCP connection with the broker. Through the broker, messages are being broadcasted across MQTT network. When a device is subscribed a specific message, it will receive all incoming messages with that topic. MQTT is very dynamic protocol, it allows to easily add and remove devices, have multiple devices listening to a single message and works easily from within most networks. For the reason’s mentioned above, and its high speed and low overhead, it has been the protocol of choice for the Internet of Things.

# Using MQTT mobile apps

To join our MQTT network, first download one of many free MQTT mobile apps for android and IOS. Once the app is installed, you will be prompted to select the broker of MQTT network. Following is our lab’s MQTT broker information: (sensitive information is being shared)   
Broker address: aerlab.ddns.net  
Port: 1883   
User name: aerlab  
Password: server

After connection to the broker, create a widget and simply select the topic that you want to subscribe or post to. List of all topics for every sensor system will be provided.  
Note: please do not post to sensor data topics, or certain control topics. Use caution controlling the Heat Dissipater.