**MQTT (Message Queuing Telemetry Transport)** là một giao thức gởi dạng publish/subscribe sử dụng cho các thiết bị [Internet of Things](/tags/IoT) với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định.

Giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng M2M (machine to machine)

**Publish, subscribe**

Trong một hệ thống sử dụng giao thức MQTT, nhiều node trạm (gọi là mqtt client - gọi tắt là client) kết nối tới một MQTT server (gọi là broker). Mỗi client sẽ đăng ký một vài kênh (topic), ví dụ như "/client1/channel1", "/client1/channel2". Quá trình đăng ký này gọi là **"subscribe"**, giống như chúng ta đăng ký nhận tin trên một kênh Youtube vậy. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gởi dữ liệu và kênh đã đăng ký. Khi một client gởi dữ liệu tới kênh đó, gọi là **"publish"**.

### QoS

Ở đây có 3 tuỳ chọn **QoS (Qualities of service)** khi "publish" và "subscribe":

Khái niệm này ra đời do nhu cầu đảm bảo sự chắc chắn trong việc gửi - nhận message giữa client và broker

* MQTT hỗ trợ 3 mức QoS
* **QoS-0** là mức đảm bảo thấp nhất, tất cả các message có QoS 0 sau khi được gửi đi bởi publisher sẽ không được kiểm tra xem đã đến broker hay chưa (fire - and - forget)
* **QoS-1**: message được đảm bảo rằng đã đến nơi nhận ít nhất 1 lần (tức là sự trùng lặp vẫn có thể xảy ra)
* **QoS-2**: đây là mức đảm bảo cao nhất, broker sẽ đảm bảo các message có QoS-2 sẽ đến nơi nhận chỉ 1 lần duy nhất, không trùng lặp, không thất lạc. Tất nhiên việc xác nhận với QoS-2 sẽ tốn băng thông hơn 2 cách còn lại.
* **Xem thêm QoS**: <https://code.google.com/p/mqtt4erl/wiki/QualityOfServiceUseCases>
* Một gói tin có thể được gởi ở bất kỳ QoS nào, và các client cũng có thể subscribe với bất kỳ yêu cầu QoS nào. Có nghĩa là client sẽ lựa chọn QoS tối đa mà nó có để nhận tin. Ví dụ, nếu 1 gói dữ liệu được publish với QoS2, và client subscribe với QoS0, thì gói dữ liệu được nhận về client này sẽ được broker gởi với QoS0, và 1 client khác đăng ký cùng kênh này với QoS 2, thì nó sẽ được Broker gởi dữ liệu với QoS2.
* Một ví dụ khác, nếu 1 client subscribe với QoS2 và gói dữ liệu gởi vào kênh đó publish với QoS0 thì client đó sẽ được Broker gởi dữ liệu với QoS0. QoS càng cao thì càng đáng tin cậy, đồng thời độ trễ và băng thông đòi hỏi cũng cao hơn.

### Retain

Nếu RETAIN được set bằng 1, khi gói tin được publish từ Client, Broker **PHẢI** lưu trữ lại gói tin với QoS, và nó sẽ được gởi đến bất kỳ Client nào subscribe cùng kênh trong tương lai.

Retain là một cờ (flag) được gắn cho một message của giao thức MQTT. Retain chỉ nhận giá trị 0 hoặc 1 (tương ứng 2 giá trị logic false hoặc true). Nếu retain = 1, broker sẽ lưu lại message cuối cùng của 1 topic kèm theo mức QoS tương ứng. Khi client bắt đầu subscribe topic có message được lưu lại đó, client ngay lập tức nhận được message.

### LWT

Gói tin LWT (last will and testament) không thực sự biết được Client có trực tuyến hay không, cái này do gói tin KeepAlive đảm nhận. Tuy nhiên gói tin LWT như là thông tin điều gì sẽ xảy đến sau khi thiết bị ngoại tuyến.

**Ngắn gọn**

Ngoài việc đóng kết nối của Client đã ngoại tuyến, gói tin LWT có thể được định nghĩa trước và được gởi bởi Broker tới kênh nào đó khi thiết bị đăng ký LWT ngoại tuyến.

<https://techmaster.vn/posts/34394/iot-giao-thuc-mqtt-va-ung-dung-trong-iot>

**MQTT Bridge**

MQTT Bridge là một tính năng của MQTT Broker cho phép các MQTT Broker có thể kết nối và trao đổi dữ liệu với nhau. Để sử dụng tính năng này, ta cần tối thiểu 2 Broker, trong đó, một Broker bất kỳ sẽ được cấu hình thành Bridge. Khi cấu hình MQTT bridge, ta cần lưu ý tới các thông số sau:

* **address**: địa chỉ của broker cần kết nối
* **bridge\_protocol\_version**: phiên bản của giao thức MQTT đang sử dụng chung cho 2 broker
* **topic**: phần này định nghĩa 3 thong số: tên topic được trao đổi giữa 2 broker, chiều trao đổi (1 chiều hay 2 chiều) và topic mapping giữa 2 broker

<http://linhkienbachkhoa.com/tutorials/esp8266-va-iot-part-2-esp8266-voi-mqtt/>

http://broker.mqtt-dashboard.com/