Message Queuing Telemetry Transport (MQTT)

Nabil Abdennadher, Raoul Dupuis,
Why MQTT and Why HTTP in not enough?

- HTTP is not enough in IoT context
  - A request/response “scenario” is not enough
  - In IoT, we:
    - have a huge number of sources of information
    - need to emit information one to many
    - need to push information over unreliable networks
In IoT, we need:

- low energy devices
- Responsiveness (near-real time delivery of information)
- Reliable delivery over non reliable connections
- Scalability
- Security and privacy (this is not unique to IoT)
MQTT @ glance

- Machine-to-machine (M2M)/"Internet of Things" connectivity protocol
- Designed as an extremely lightweight publish/subscribe messaging transport.
- Useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium
- Invented by Dr. Andy Stanford-Clark of IBM and Arlen Nipper of Arcom (now Eurotech) in 1999
- OASIS (Organization for the Advancement of Structured Information Standards) standard
- ISO standard (ISO/IEC PRF 20922)
- Used by Amazon Web Services, IBM WebSphere MQ, Microsoft Azure IoT, Facebook Messenger

http://mqtt.org/
MQTT

Publisher  Broker  Subscriber
Topics

• A producer sends (publishes) a message (publication) on a topic (subject)
• A consumer subscribes (makes a subscription) for messages on a topic (subject)
• The broker matches publications to subscriptions
  • If no matches the message is discarded
  • If one or more matches the message is delivered to each matching subscriber/consumer
• Ex: CH/Geneva/1214/Abdennadher/energyConsumption (same for solar, alarm)
• Single-level wildcards “+” can appear anywhere in the topic string
  • CH/Geneva/1214/+/energyConsumption – Energy consumption for all houses in Vernier (Postal Code: 1214)
• Multi-level wildcards “#” must appear at the end of the string
  • Ex: CH/Geneva/1214/Abdennadher/# (energy consumption, solar and alarm)
• Wildcards must be next to a separator
• Cannot use wildcards when publishing
Features

• Small code footprint
• Ideal if processor or memory resources are limited
• Ideal if bandwidth is low or network is unreliable
• Publish/subscribe message exchange pattern
• Works on top of TCP/IP, but also other protocols (ex: Zigbee)
• Security: authentication using user name and password, encryption using SSL/TLS
• Persistence: MQTT has support for persistent messages stored on the broker.
• MQTT over WebSocket possible (browser as MQTT client)
MQTT vs. WebSocket

- HTTP, WebSocket and MQTT are messaging protocols (Application Layer)
- WebSocket defines how can you raise a HTTP connection into a bidirectional channel
- In a higher level, MQTT defines how several endpoint exchange data when they can talk with the same broker.
- MQTT works on WebSocket, TCP.
- If we build an IoT system, the probably architecture is we use MQTT over TCP in the connected devices, and we use MQTT over WebSocket in the web pages.