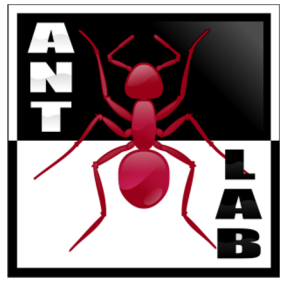
Politecnico di Milano

Advanced **N**etwork **T**echnologies **Lab**oratory



Internet of Things

MQTT

Agenda

- MQTT recap
- MQTT command line
- MQTT packet sniffing

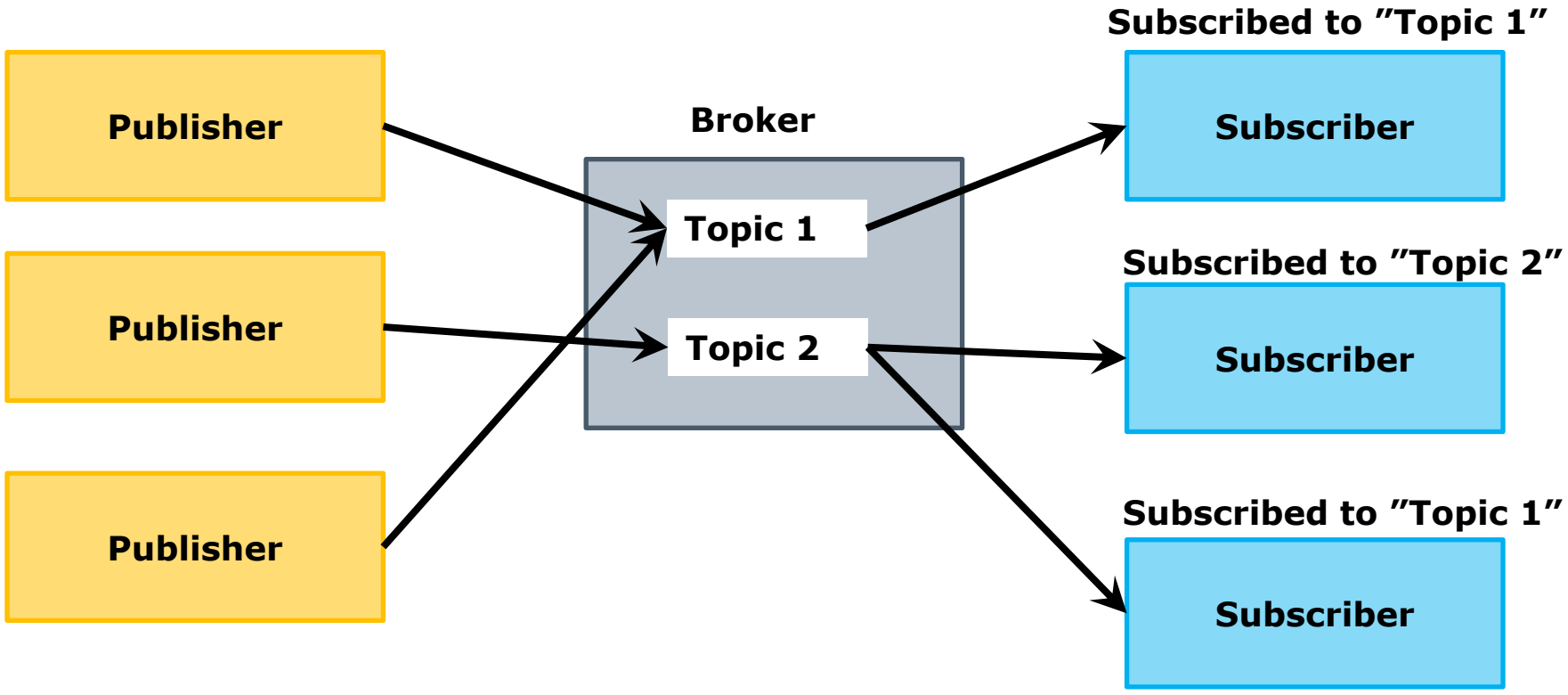
What we'll use

- Virtual Machine
- Command line
- Mosquitto broker/clients
- Wireshark
- Your brainz

MQTT in a nutshell

- Publish/subscribe protocol
- Easy client-side implementation
- Protocol optimized for unreliable, low-bandwidth, high-latency networks
- Data agnostic
- Few methods: **publish**/**subscribe**/unsubscribe

MQTT pub-sub model



MQTT Actors

- One Broker
- Many clients
  - Publishers
  - Subscribers

MQTT Broker on the cloud

- Address:
  - mqtt.eclipse.org
  - test.mosquitto.org
  - broker.hivemq.com
- Port:
  - 1883
  - 8883

https://github.com/mqtt/mqtt.github.io/wiki/public_brokers

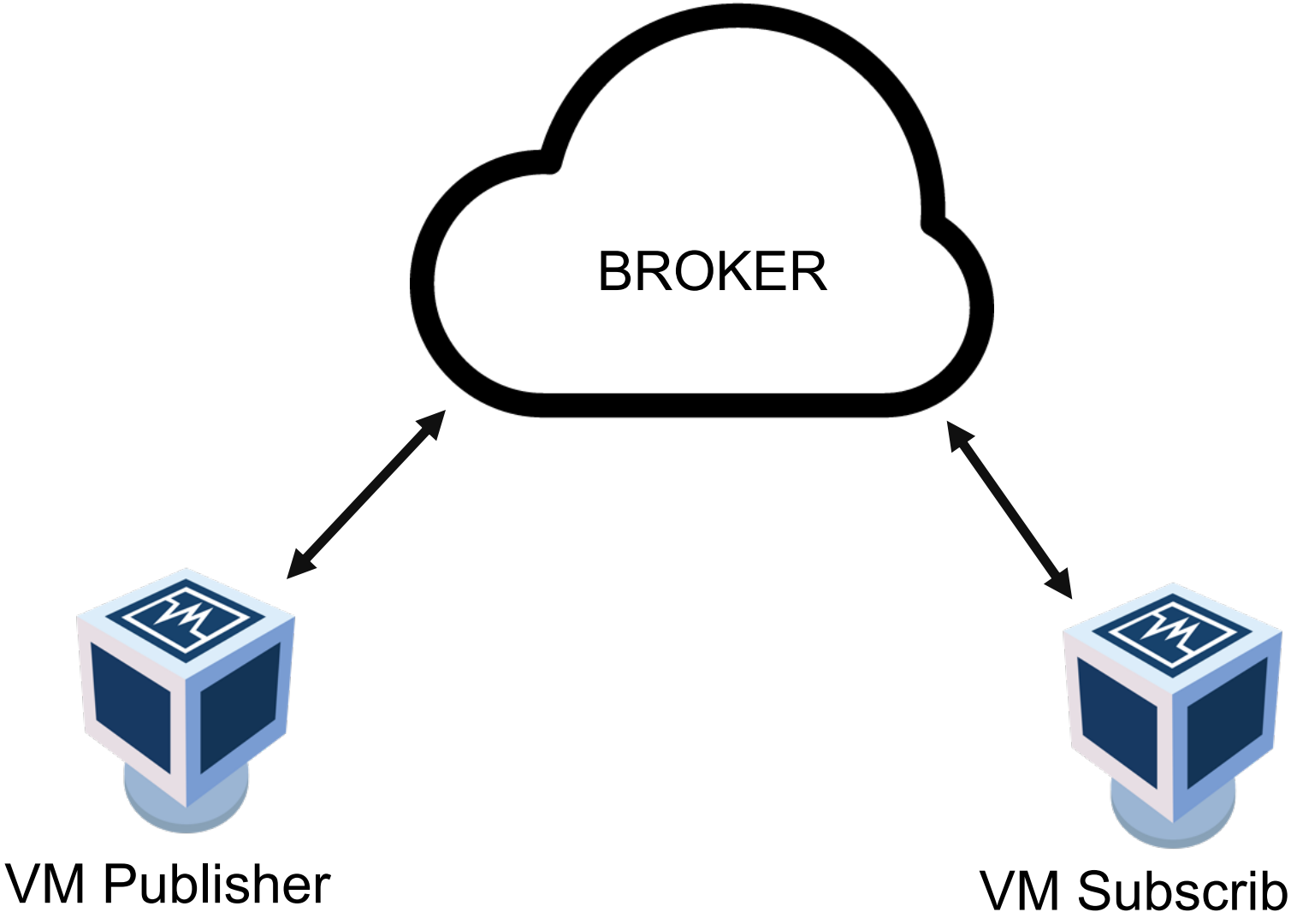
Mosquitto clients

- Publish a message: mosquitto_pub
  - -h “host_name”
  - -p “port”
  - -t “topic_name”
  - -m “message_content”
  - -q “qos”
  - -d “debug”

Mosquitto clients

- Subscribe to a topic: mosquitto_sub
  - -h “host_name”
  - -p “port”
  - -t “topic_name”
  - -q “qos”
  - -d ”debug”
  - -v “verbose”

Broker on the cloud



Topic

- String that the broker uses to filter messages
- Identify a resource

```
building20/floor1/room1.2/temperature
```

Wildcards

- Example of topics
  - Topic #1: `home/groundfloor/kitchen/temperature`
  - Topic #2: `home/groundfloor/living_room/luminance`
- Wildcards
  - Single-level: `home/groundfloor/+/temperature`

  (to subscribe to **all the temperature reading** in all the room of the ground floor)
  - Multi-level: `home/groundfloor/#`

  (to subscribe to **all the readings** in the ground floor, **not only the temperature in the living room**)

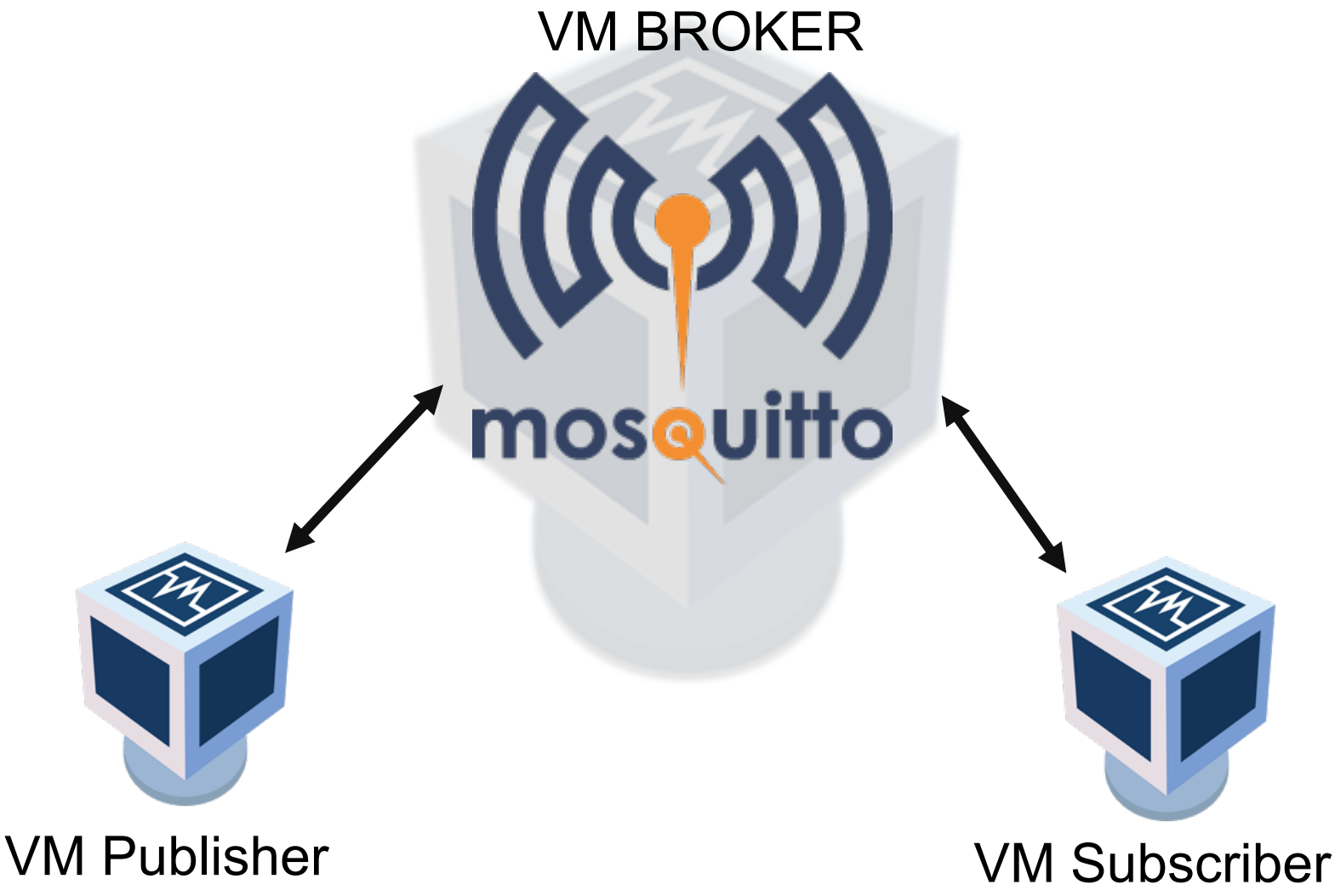
Wildcards

- Single level: +
  - Can be used in the middle of the topic
  - como/+/temperature :
    - deib/eg1/temperature YES
    - deib/L26.1/temperature YES
    - deib/eg1/humidity NO!
- Multi level: #
  - Used only at the end of the topic
  - deib/eg1/# :
    - deib/eg1/temperature YES
    - deib/eg1/humidity YES
    - deib/L26.1/temperature NO!

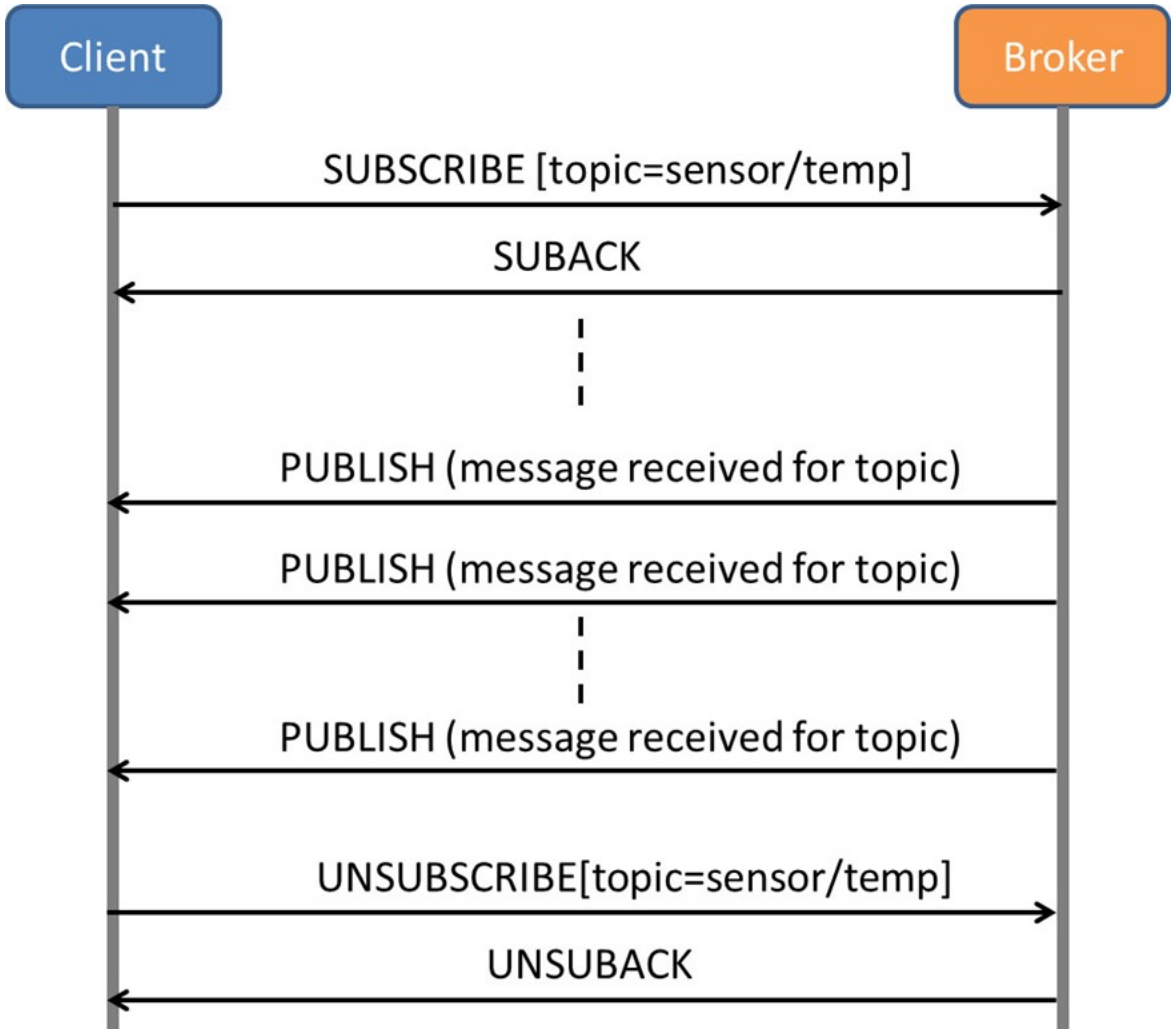
MQTT LOCAL BROKER

- Start the broker:
  - `mosquitto [-d | --daemon ][-p` *`port number`*`][-v | --verbose]`
- Example:
  - `mosquitto -p 1883 -v`

Broker on the VM



MQTT Sub-Unsubscribe

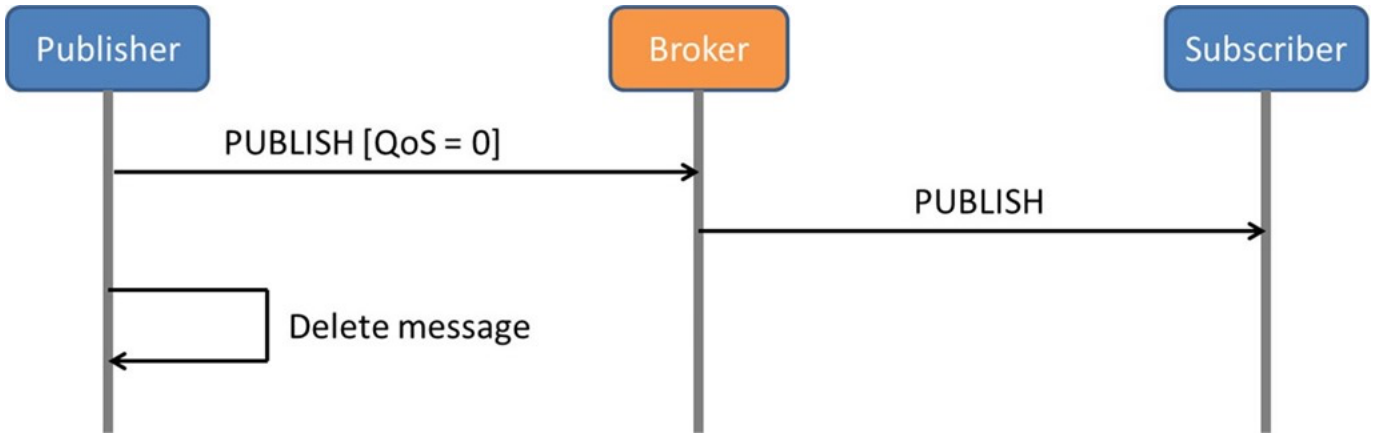


http://www.slideshare.net/paolopat/mqtt-iot-protocols-comparison

QoS

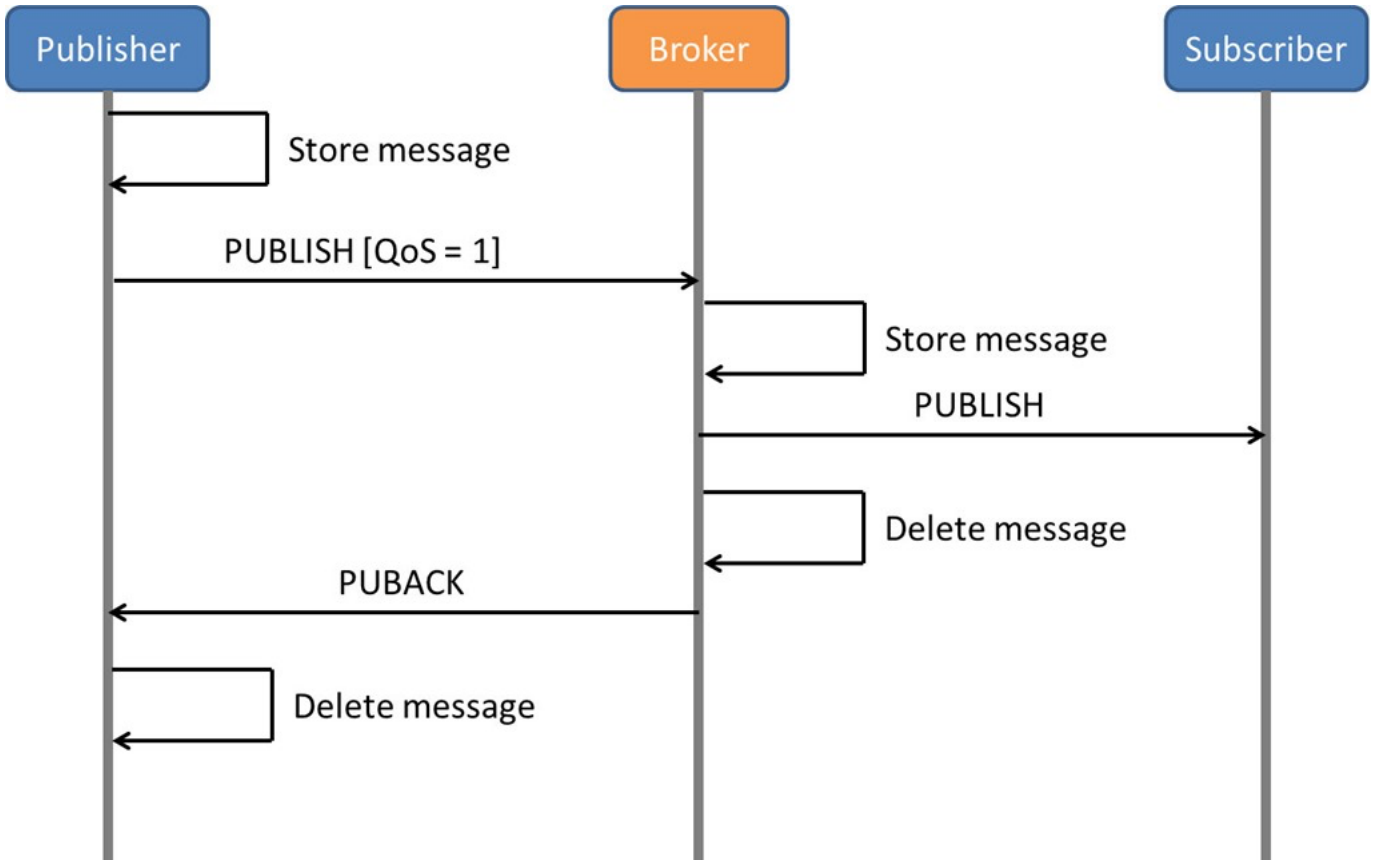
- QoS 0: at most once
  - Doesn’t survive to failure
  - No duplicates
- QoS 1: at least once
  - Survives connection loss
  - Duplicates
- QoS 2
  - Survives connection loss
  - No Duplicates

Publisher/sub QOS 0



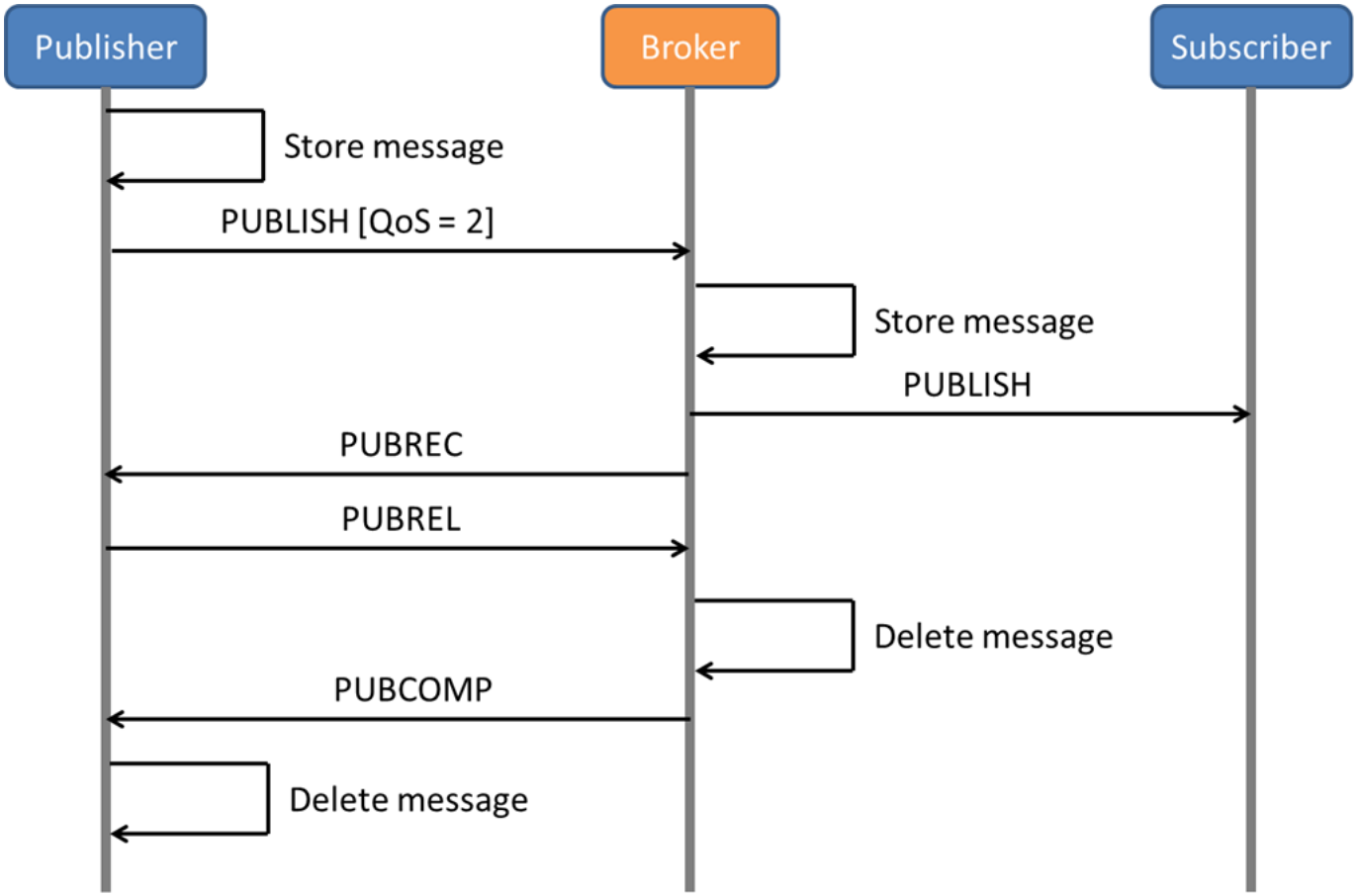
http://www.slideshare.net/paolopat/mqtt-iot-protocols-comparison

Publisher QOS 1



http://www.slideshare.net/paolopat/mqtt-iot-protocols-comparison

Publisher QOS 2



http://www.slideshare.net/paolopat/mqtt-iot-protocols-comparison

Retain message

- Messages are normally discarded by the broker if no one is subscribed to that topic
- With the retain flag the broker save the last message on that topic
- When a subscriber subscriber on the topic, the broker deliver the message

Last Will message

- Used to notify subs of an unexpected shut down of the publisher
- When the broker detects a connection break it sends the last will msg to all subs of a topic
- Normal disconnect: NO msgs
- Abnormal disconnect: Last Will

$SYS topics

- Topics created by the broker to keep track of the broker's status
- Starts with $SYS
- Automatic periodic publish
- Some example:
  - **$SYS/broker/clients/connected**
  - **$SYS/broker/messages/received**
  - **$SYS/broker/uptime**

https://github.com/mqtt/mqtt.github.io/wiki/SYS-Topics

Online broker and clients

- Public available brokers:
  - test.mosquitto.org
  - broker.hivemq.com
  - https://github.com/mqtt/mqtt.github.io/wiki/public_brokers
- Online clients:
  - http://www.hivemq.com/demos/websocket-client/
  - https://mqttboard.flespi.io/#/
  - https://github.com/mqtt/mqtt.github.io/wiki/tools

REST vs MQTT

	CoAP	MQTT
Model used for communication	Request-Response, Publish-Subscribe	Publish-Subscribe
RESTful	Yes	No
Transport layer	Preferably UDP, TCP can also be used.	Preferably TCP, UDP can also be used (MQTT-S).
Messaging	Asynchronous & Synchronous	Asynchronous
Application Reliability	2 QoS	3 QoS
Application success stories	Utility Field Area Networks	Extending enterprise messaging into IoT applications
Paradigm	One-to-One	Many-to-many

Recap of today

- How to start mosquitto broker
- Publish a message
- Subscribe to a topic
- Subscribe to + wildcard
- Subscribe to # wildcard

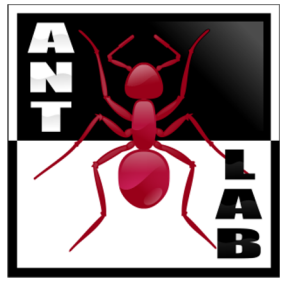
Recap of today

- Connection message exchange
- Subscribe message exchange
- QoS 0,1,2
- Retain message
- Will message
- Broker-client ping
- $SYS topic

https://github.com/edoardesd/iot-examples

Politecnico di Milano

Advanced **N**etwork **T**echnologies **Lab**oratory

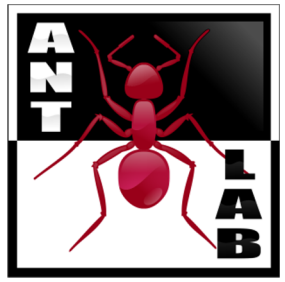


Wireshark

MQTT packet sniffing

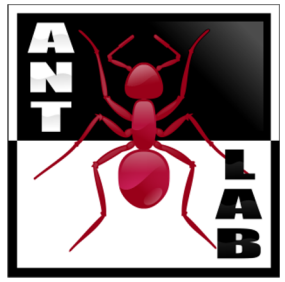
Politecnico di Milano

Advanced **N**etwork **T**echnologies **Lab**oratory

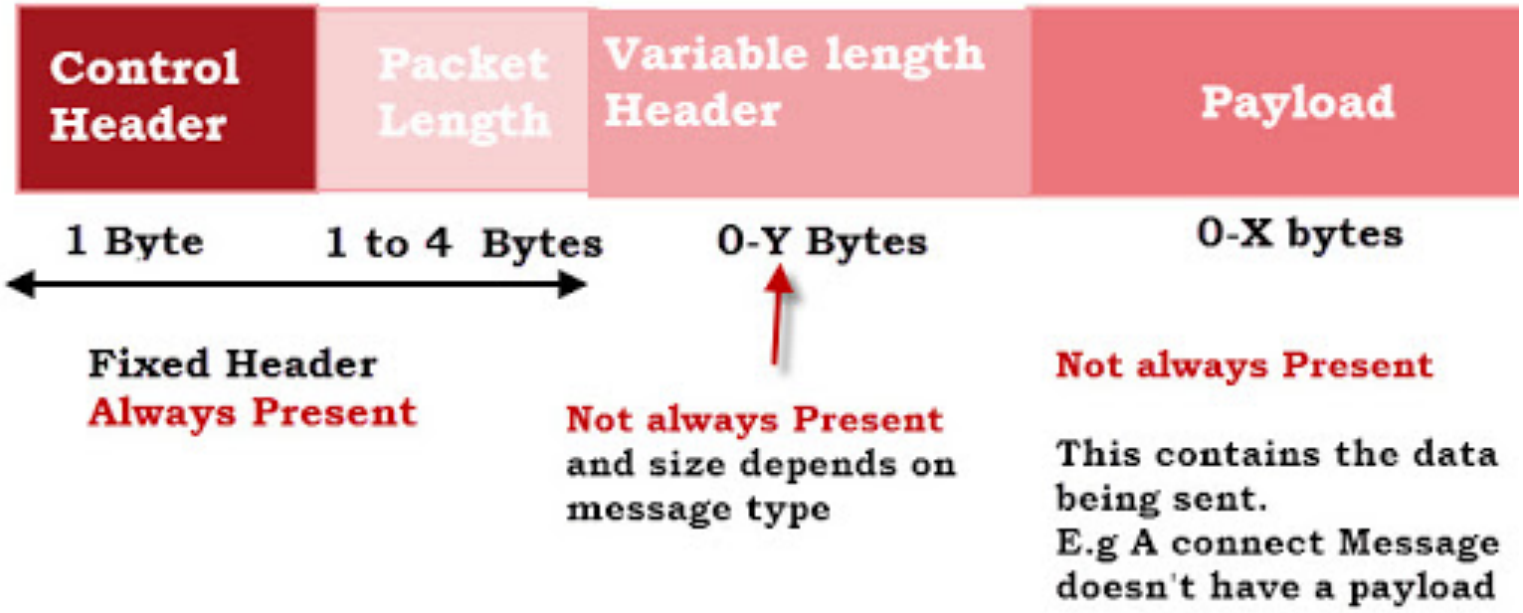


Politecnico di Milano

Advanced **N**etwork **T**echnologies **Lab**oratory



MQTT packet structure



MQTT Standard Packet Structure

From: steve's internet guide

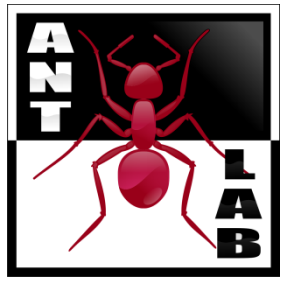
Wireshark useful filters

- mqtt.clientid
- mqtt.kalive
- mqtt.len
- mqtt.qos
- mqtt.retain
- mqtt.topic
- mqtt.topic_len

- For a complete reference use the documentation:
  https://www.wireshark.org/docs/dfref/m/mqtt.html

Parse the pcap (advanced)

- Python: https://scapy.readthedocs.io/en/latest/usage.html
- Java: https://formats.kaitai.io/pcap/java.html
- C++: https://pcapplusplus.github.io
- Javascript: https://www.npmjs.com/package/pcap-parser
- …



Challenge

Home challenge #3: Sniffing

Home Challenge #3

- Download the homework3.pcap file from the beep
- Analyse the traffic with wireshark or with something else

- Answer the following questions

Questions

1) What's the difference between the message with MID: 3978 and the one with MID: 22636?

2) Does the client receive the response of message No. 6949?

3) How many replies of type connectable and result code "Content" are received by the server "localhost"?

4) How many messages with the topic "factory/department/+" are published by a client with user name: "jane"?

5) How many clients connected to the broker "hivemq" have specified a will message?

Questions

1) How many publishes with QoS 1 don’t receive the ACK?

2) How many last will messages with QoS set to 0 are actually delivered?

3) Are all the messages with QoS > 0 published by the client “4m3DWYzWr40pce6OaBQAfk” correctly delivered to the subscribers?

4) What is the average message length of a connect msg using mqttv5 protocol? Why messages have different size?

5) Why there aren’t any REQ/RESP pings in the pcap?

Challenge deliverable

- A PDF containing motivated responses:
  - + filters and/or code, if used
- Your names + ID number/matricola

Homework Deadline

- Deadline: May 17, 2020 @23:49
- Max 3 people

- Use the folder #3 on beep