POLITECNICO
MILANO 1863

ANTLAB

Internet of Things Lab

Lab 2: MQTT

Agenda

- MQTT Recap
- MQTT in action
- MQTT Packet Sniffing
- **Challenge 1!**

What we will use

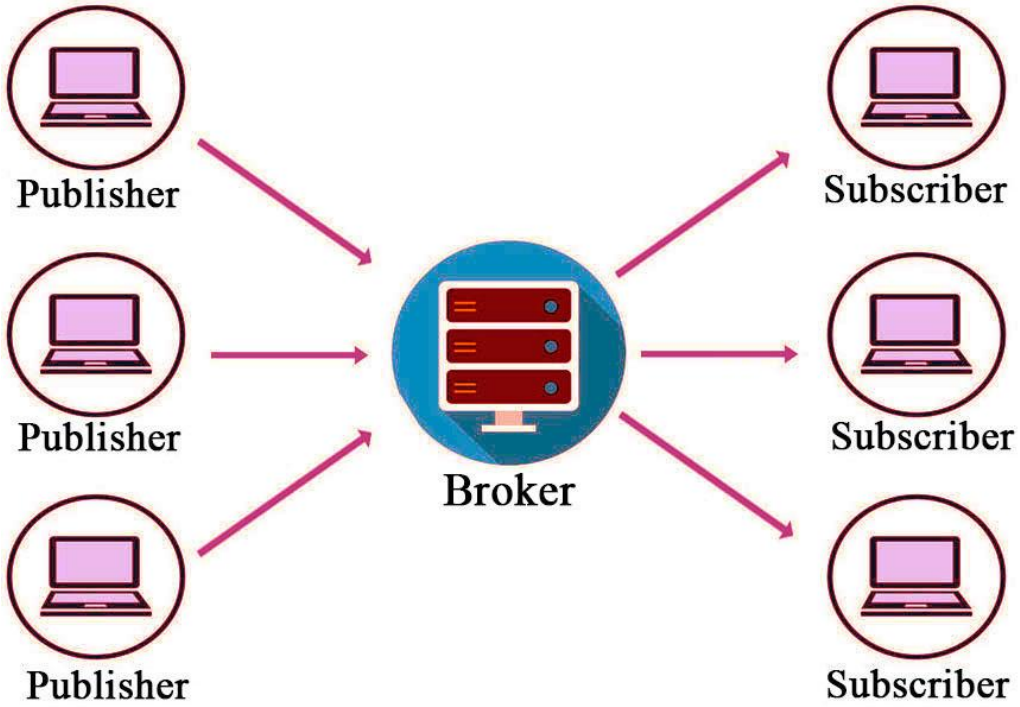
- Virtual Machine
- Terminal
- Mosquitto broker and clients
- Wireshark
- **Your attention ☺**

MQTT in a nutshell

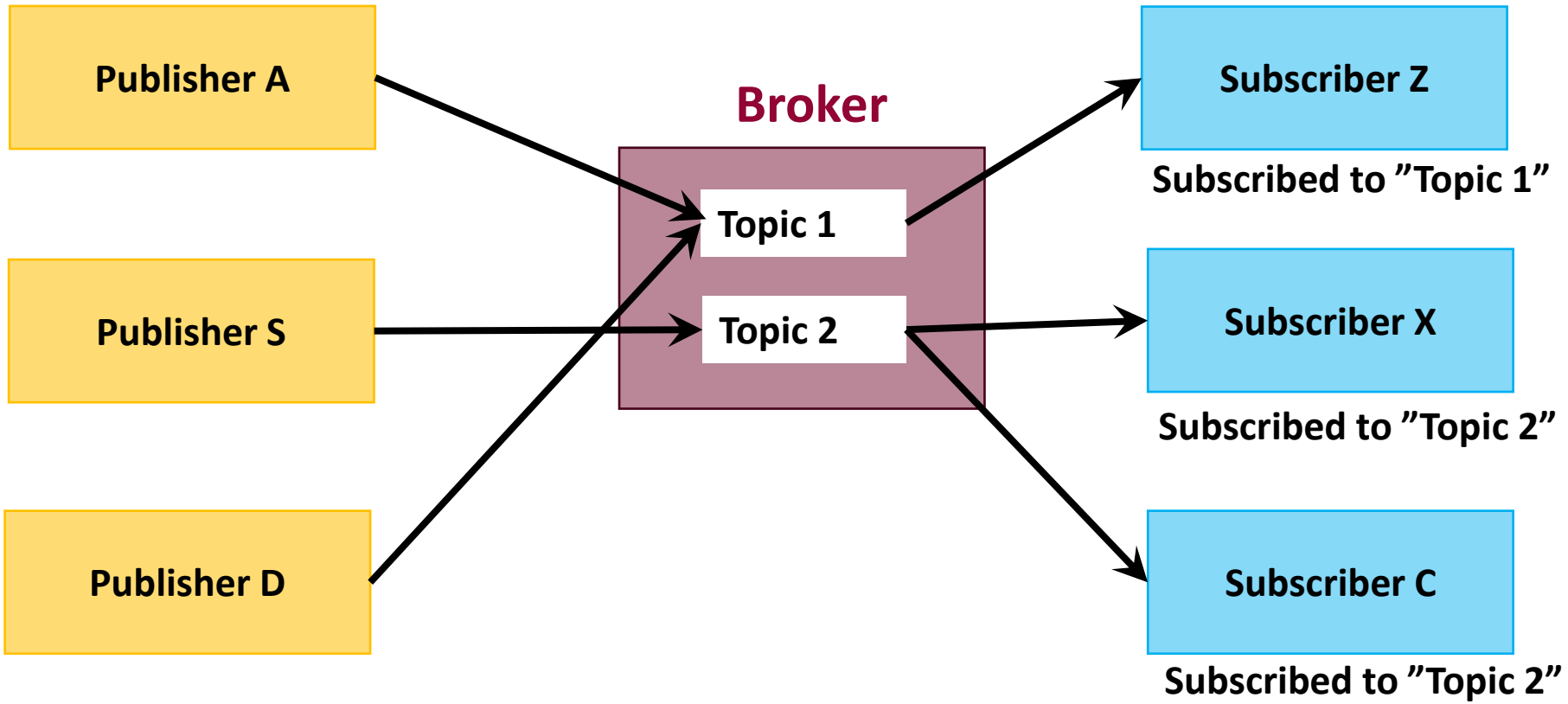
- Protocol optimized for unreliable, low-bandwidth, high-latency networks
- Lightweight
- Easy client-side implementation
- Data agnostic
- Publish/subscribe protocol
- Few methods: **publish**/**subscribe**/unsubscribe

MQTT actors

- One broker
- Many Clients
  - Publishers
  - Subscribers



MQTT pub-sub model



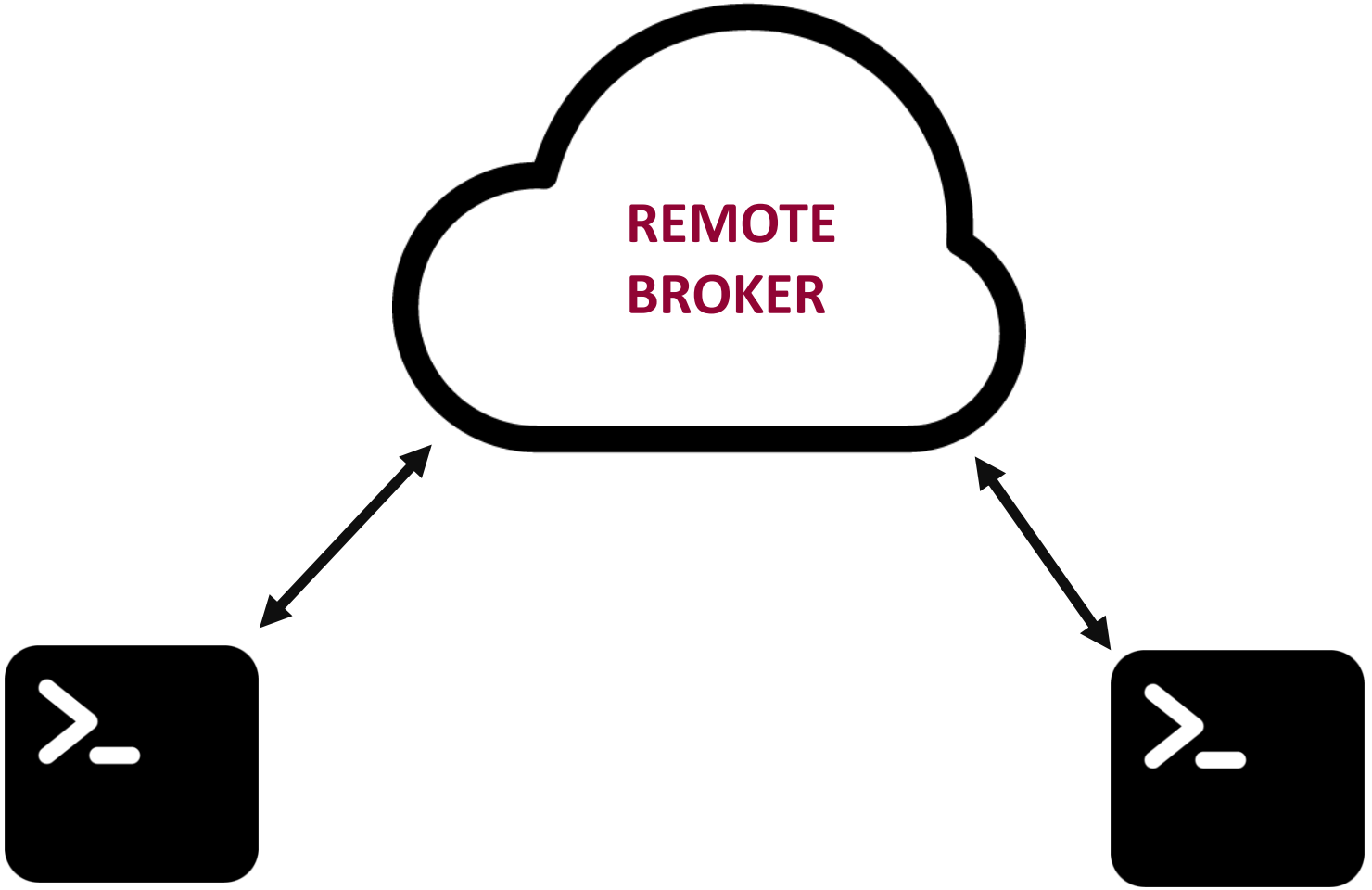
Broker on the cloud

- Addresses:
  - test.mosquitto.org
  - broker.hivemq.com

- Ports:
  - 1883
  - 8883

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

Broker on the cloud



Mosquitto clients: SUBSCRIBER

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

Example: mosquitto_sub –h localhost –t "/living_room/temperature" -v -d

Mosquitto clients: PUBLISHER

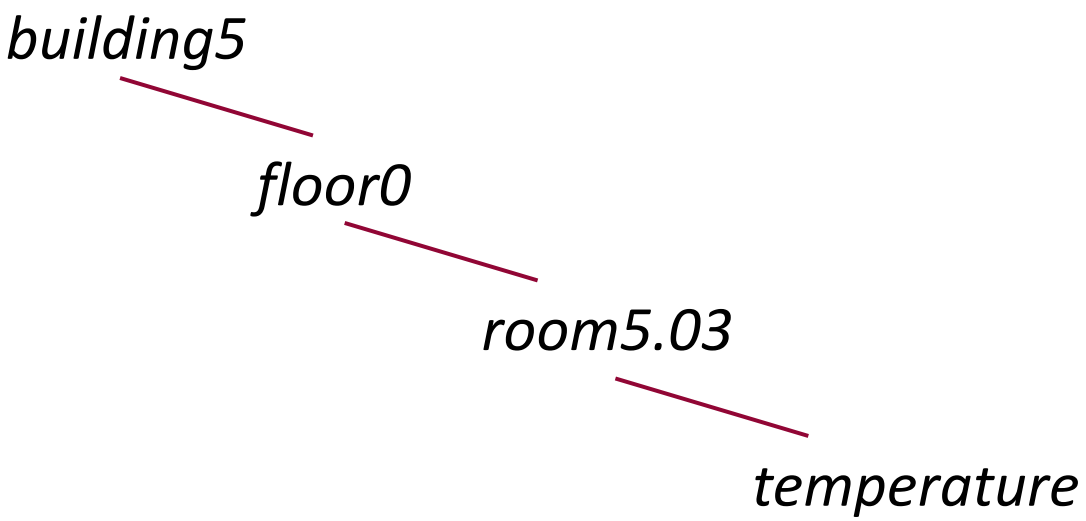
- Subscribe to a topic: **mosquitto_pub**
  - -h “host_name”
  - -p “port”
  - -t “topic_name”
  - -m “message_content”
  - -q “qos”
  - -d ”debug”
  - -r “retain”

Example: mosquitto_pub –h localhost –t "/living_room/temperature" –m "33.3" -d

Topic

- String that the broker uses to filter messages
- Identifies a resource
- Hierarchical structure

Example: *building5/floor0/room5.03/temperature*



Topic and Wildcards

- Examples of topic:
  - Topic #1: home/groundfloor/kitchen/temperature
  - Topic #2: home/groundfloor/bed_room/luminance
  - Topic #3: home/groundfloor/bed_room/temperature

- **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** of the ground floor, **not only the temperature in the living room**)

Wildcards

- Single level: **+**
  - Can be used in the middle or at end of the topic
  - Only one level

- Multi level: **#**
  - Take any number of levels
  - Used **only at the end** of the topic

Wildcards Examples

- deib/+/temperature :
    - deib/eg1/temperature **??**
    - deib/L26/room.1/temperature **??**
    - deib/eg1/humidity **??**

Wildcards Examples

- deib/+/temperature :
    - deib/eg1/temperature **YES!**
    - deib/L26/room.1/temperature **NO!**
    - deib/eg1/humidity **NO!**

Wildcards Examples

- deib/+/temperature :
    - deib/eg1/temperature **YES!**
    - deib/L26/room.1/temperature **NO!**
    - deib/eg1/humidity **NO!**

- deib/eg1/#:
    - deib/eg1/temperature **??**
    - deib/eg1/humidity **??**
    - deib/L26.1/temperature **??**

Wildcards Examples

- deib/+/temperature :
    - deib/eg1/temperature **YES!**
    - deib/L26/room.1/temperature **NO!**
    - deib/eg1/humidity **NO!**

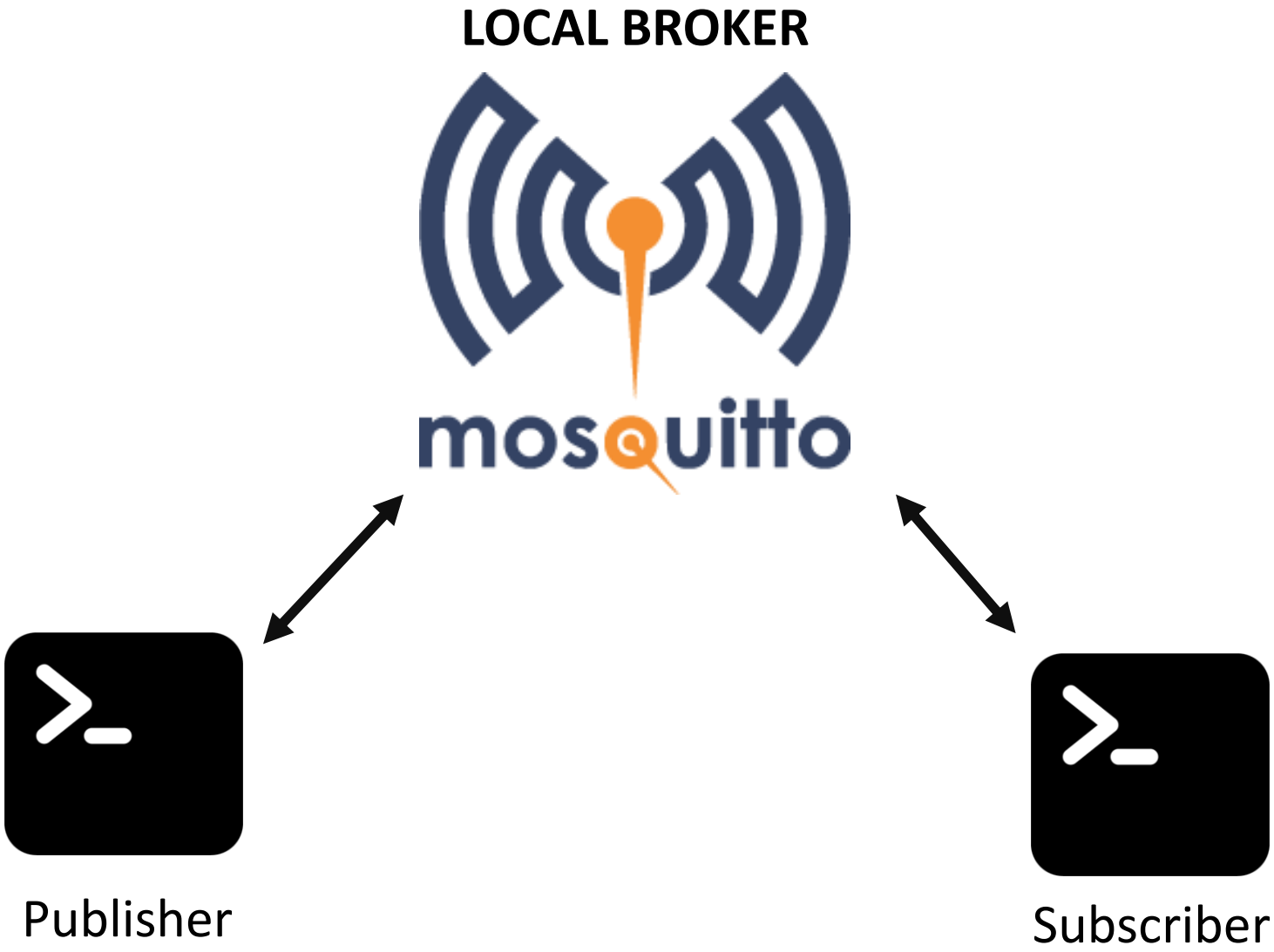
- deib/eg1/#:
    - deib/eg1/temperature **YES!**
    - deib/eg1/humidity **YES!**
    - deib/L26.1/temperature **NO!**

MQTT Local Broker

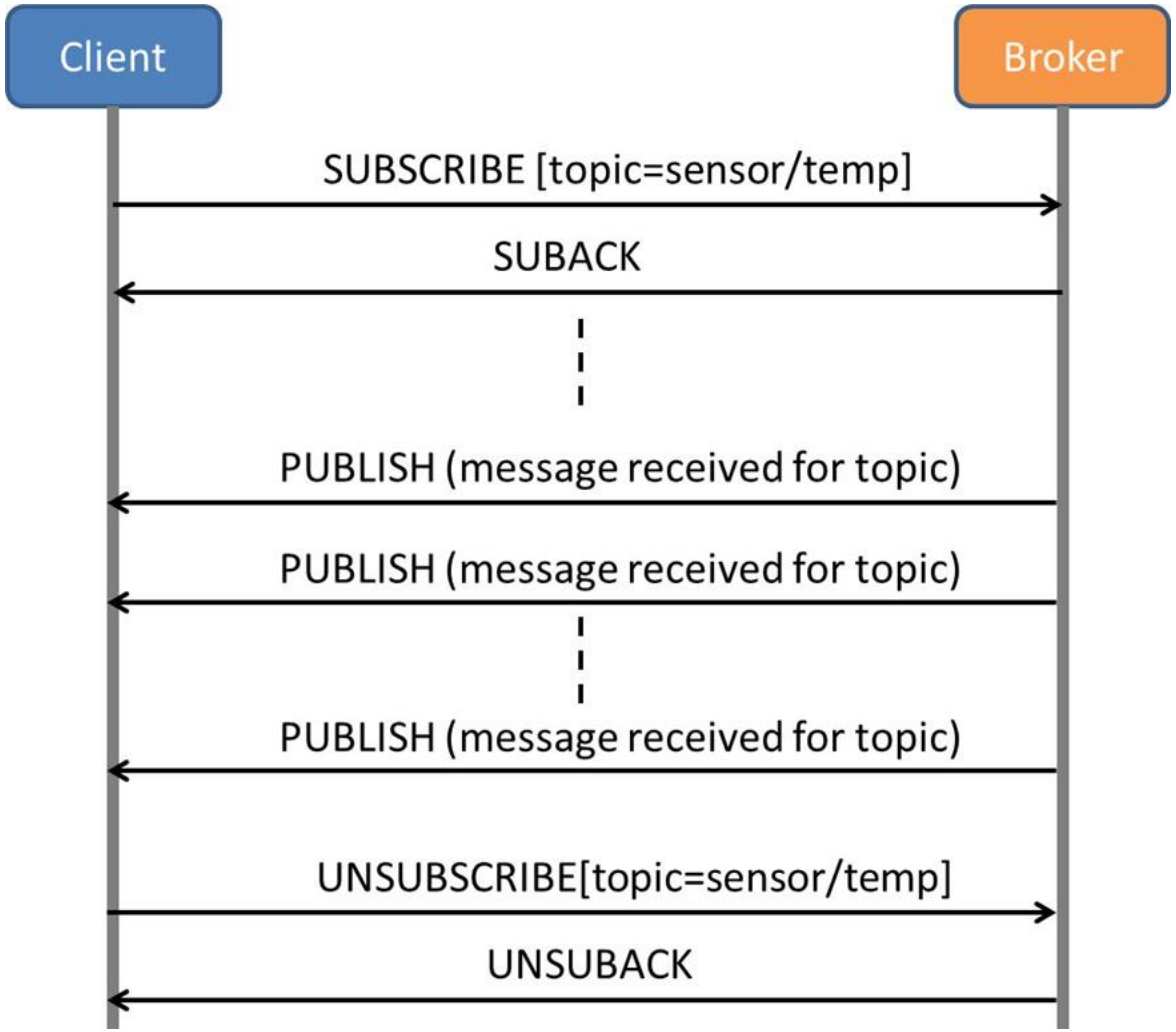
- Start the broker:
  mosquitto [-d | --daemon][-p *port number*] [-v | --verbose]

- Example:
  mosquitto –p 1883 -v

Broker on the VM



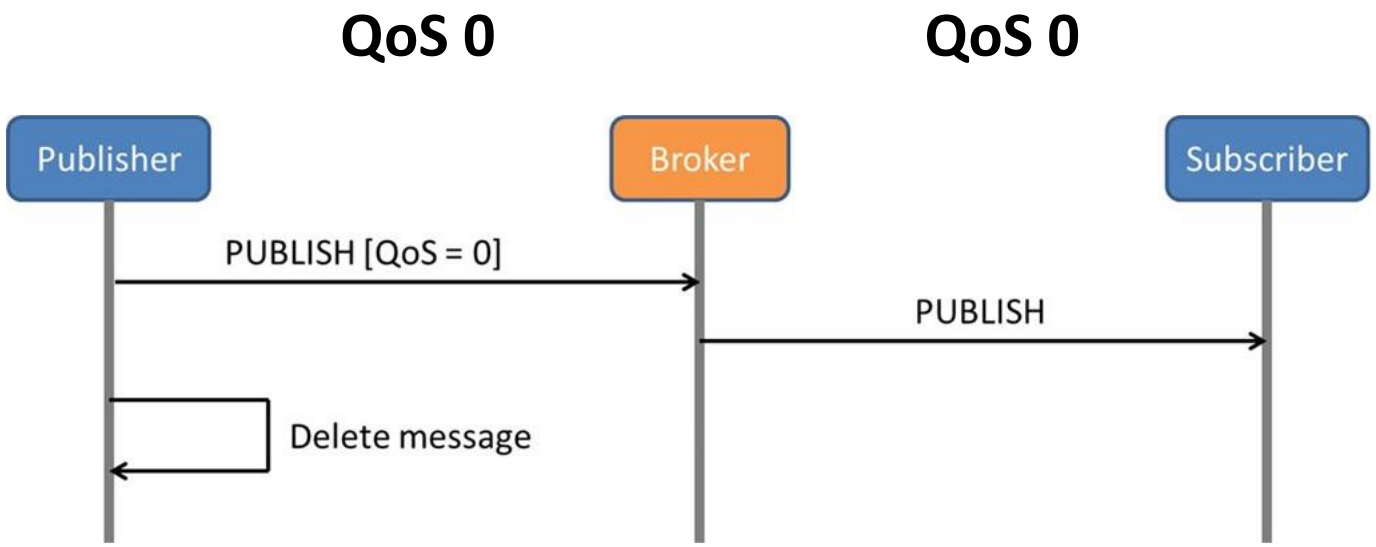
MQTT Subscribe/Unsubscribe



QoS

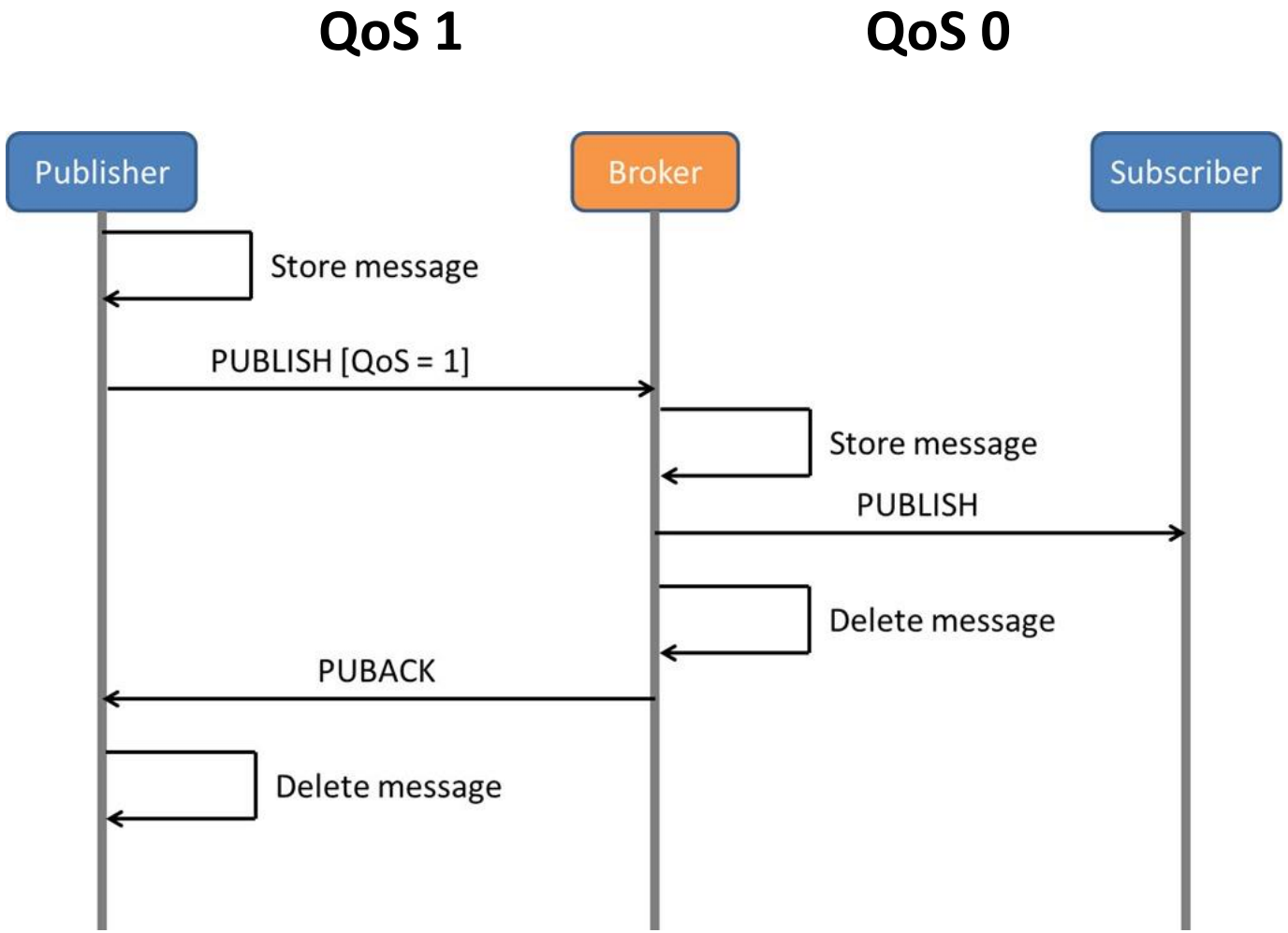
- **QoS 0**: at most once
  - Doesn’t survive to failure
  - No duplicates
- **QoS 1**: at least once
  - Survives connection loss
  - Duplicates
- **QoS 2**: exactly once
  - Survives connection loss
  - No Duplicates

Publisher with QoS 0



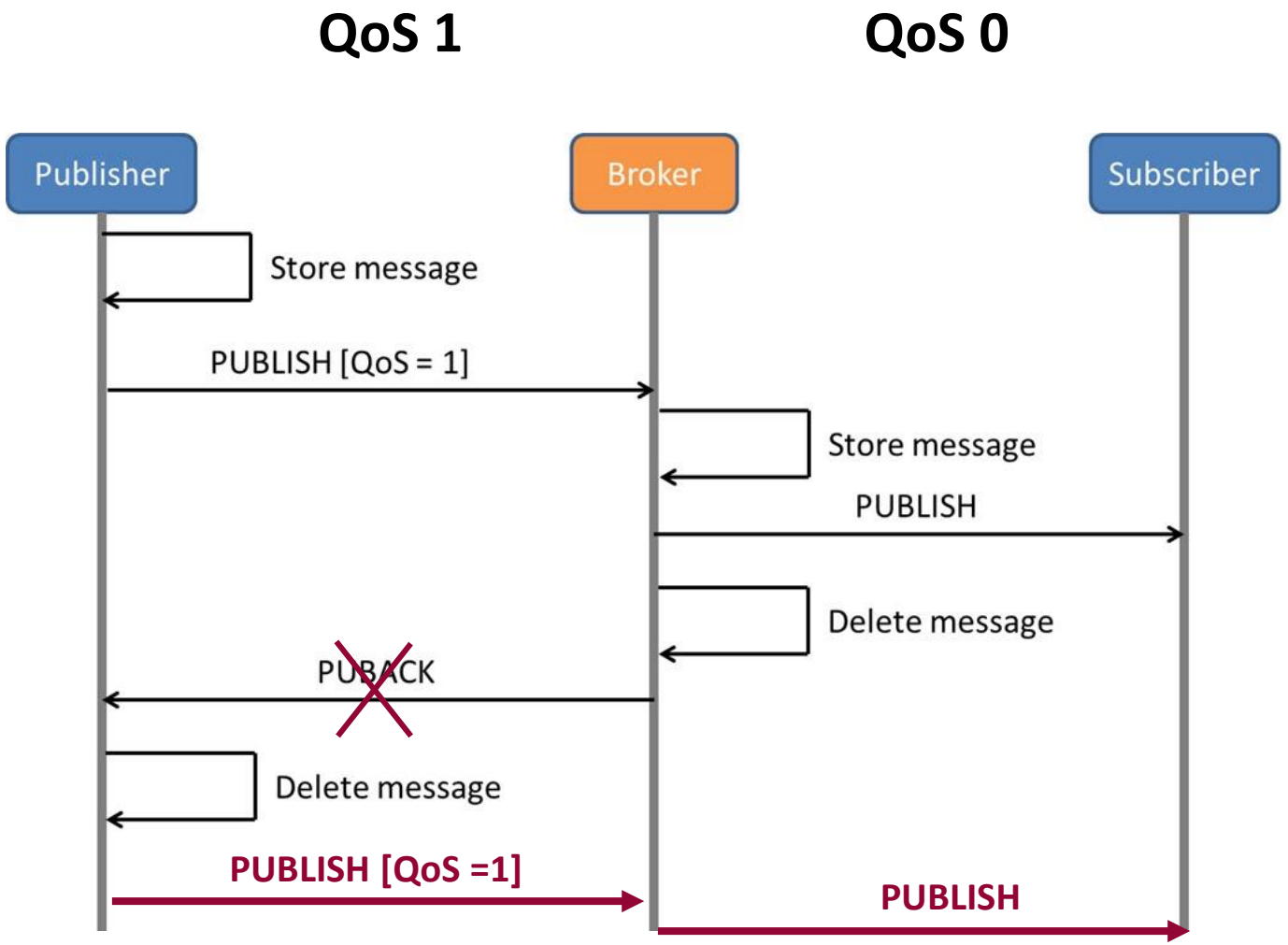
http://www.slideshare.net/paolopat/mqtt-iot-protocols-comparison

Publisher with QoS 1



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

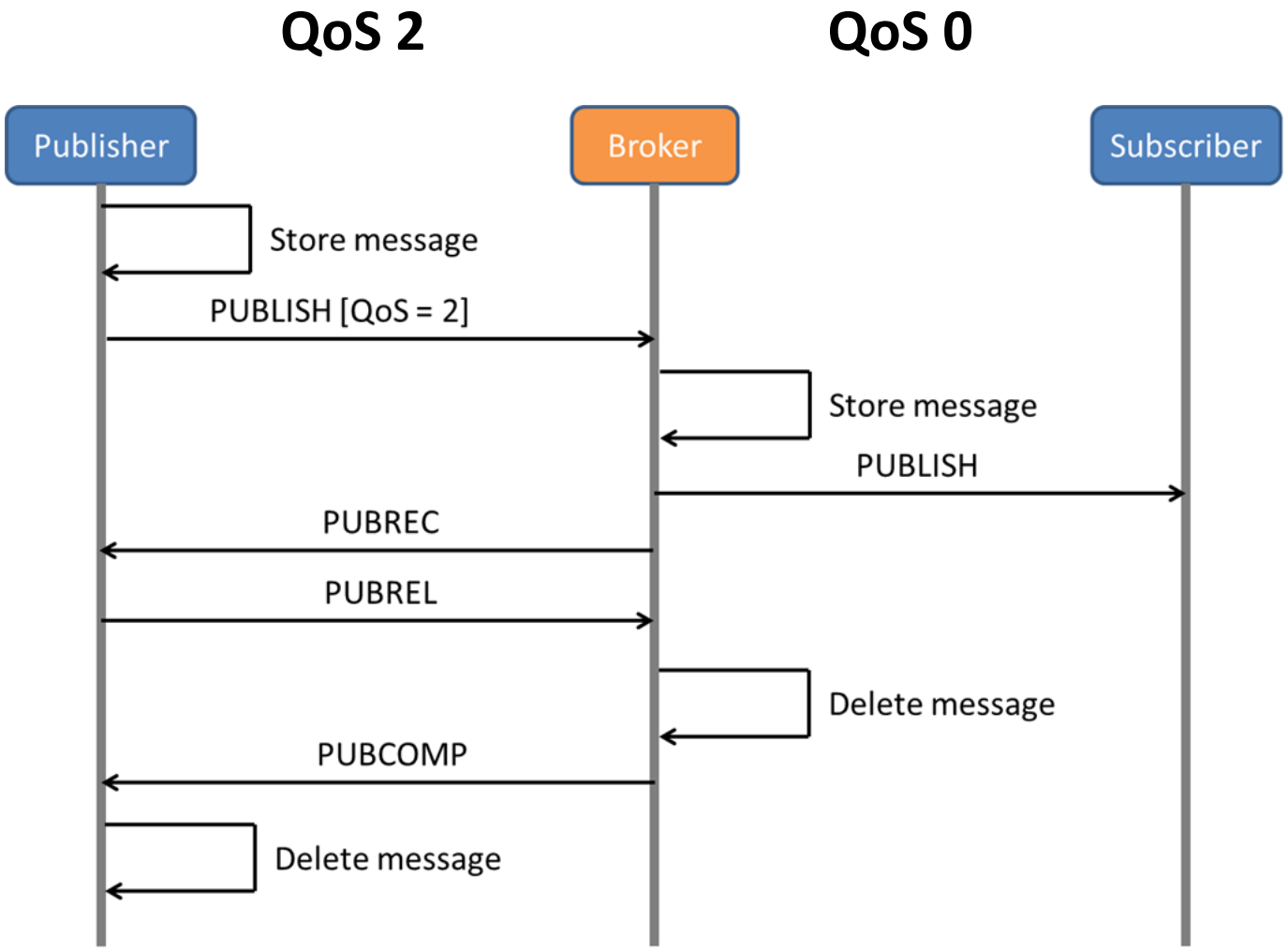
Publisher with QoS 1 (PUBACK LOSS)



DUPLICATED!

ANTLAB

Publisher with QoS 2



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

ANTLAB

QoS Recap

- Subscribers can choose the **MAXIMUM** QoS value to support.
- The resulting QoS for the sub is the **minimum** value between QoS_{SUB} and QoS_{PUB}

Publisher QoS	Subscriber QoS	Resulting Sub QoS
0	0,1,2	0
1	0	0
1	1	1
2	0	0
2	1	1
2	2	2

Retain message

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

mosquitto_pub –t “topic” –h localhost –m “my message” -r

Persistent Session

- Persistent session stores session status and messages with QoS 1 and 2 that are still to be transmitted or acknowledge
- Setting the **clean session** flag to 0, the broker will reuse the previous session when connecting (It is mandatory to specify the client ID with *-i*

mosquitto_pub –t “topic” –h localhost –m “my message” –i “pub_cli” – q 1 – c

mosquitto_sub –t “topic” –h localhost –i “spub_cli” – q 1 – c

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: Send Last Will

$SYS topics

- Topics created by the broker to keep track of the broker's status
- Starts with $SYS
- Automatic periodic publish
- Some examples:
  - **$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

CoAP 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-SN).
Messaging	Asynchronous & Synchronous	Asynchronous
Application Reliability	2 QoS (CON, NON)	3 QoS (0,1,2)
Application success stories	Utility Field Area Networks	Extending enterprise messaging into IoT applications
Paradigm	One-to-One, Many-to-many	Many-to-many

Recap of today

- Mosquitto broker
- Publish a message
- Subscribe to a topic
- Subscribe with wildcard
- Connection management (CONNECT, PINGS)
- Different QoS values
- Retain and Last Will features
- $SYS topics

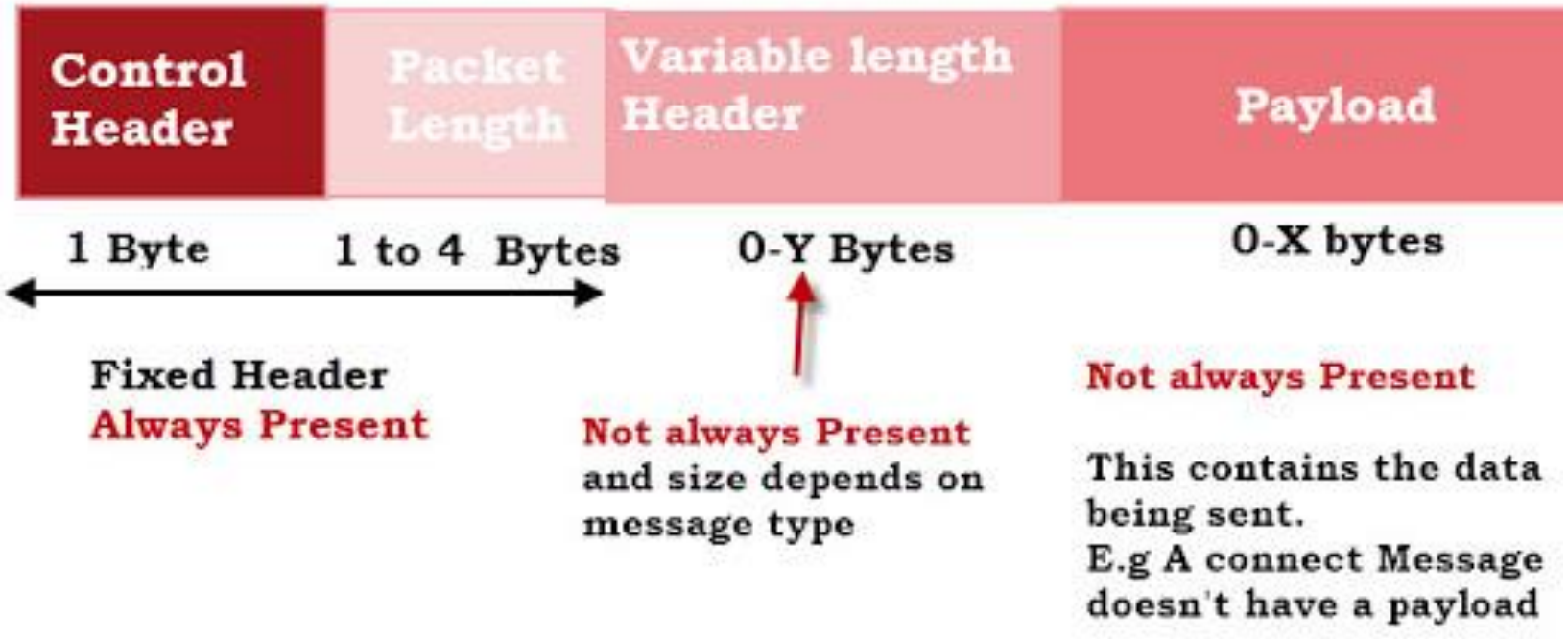
POLITECNICO
MILANO 1863

ANTLAB

Wireshark

MQTT packet sniffing

MQTT Packet structure



MQTT Standard Packet Structure

From: steve's internet guide

Wireshark MQTT filters

- mqtt.msgtype
- mqtt.clientid
- mqtt.kalive
- mqtt.len
- mqtt.qos
- mqtt.retain
- mqtt.topic
- mqtt.topic_len
- For more 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
- …